What's new in iOS 7

Spoiler alert: a lot.

Adrian Kosmaczewski

TRIFORK.

...think software

Bringing Web Programming to Mobile Devices

Mobile JavaScript Application Development

Adrian Kosmaczewski

O'REILLY®

Building Enterprise Cross-Platform Mobile Web Applications

Sencha Touch 2

Up and Running

Adrian Kosmaczewski

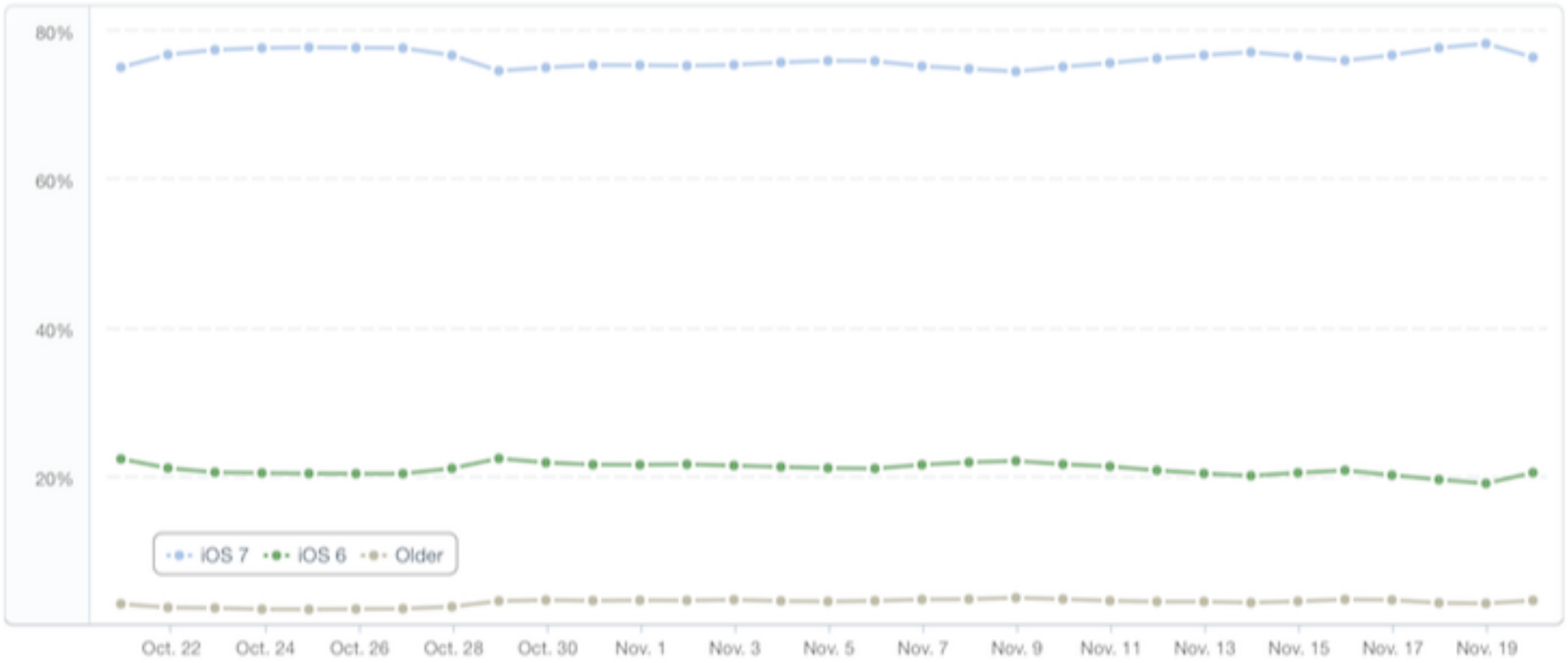
Summary

If you just stay for one slide, this is it.

New UI
Dynamic behavior
Text Kit
64-bit Support
More Multitasking
Sprite Kit
Game Controller
Inter App Audio
Peer-to-Peer Connectivity
…and more…

iOS 7 adoption

Hour | Day



Time/Date in UTC

THIS REPORT WAS GENERATED FROM **28,246,600,412** RECORDS.

Thanks for coming!

(no no no wait!)

New UI

Flatter than flat.

New UI: Key Concepts

Deference

Clarity

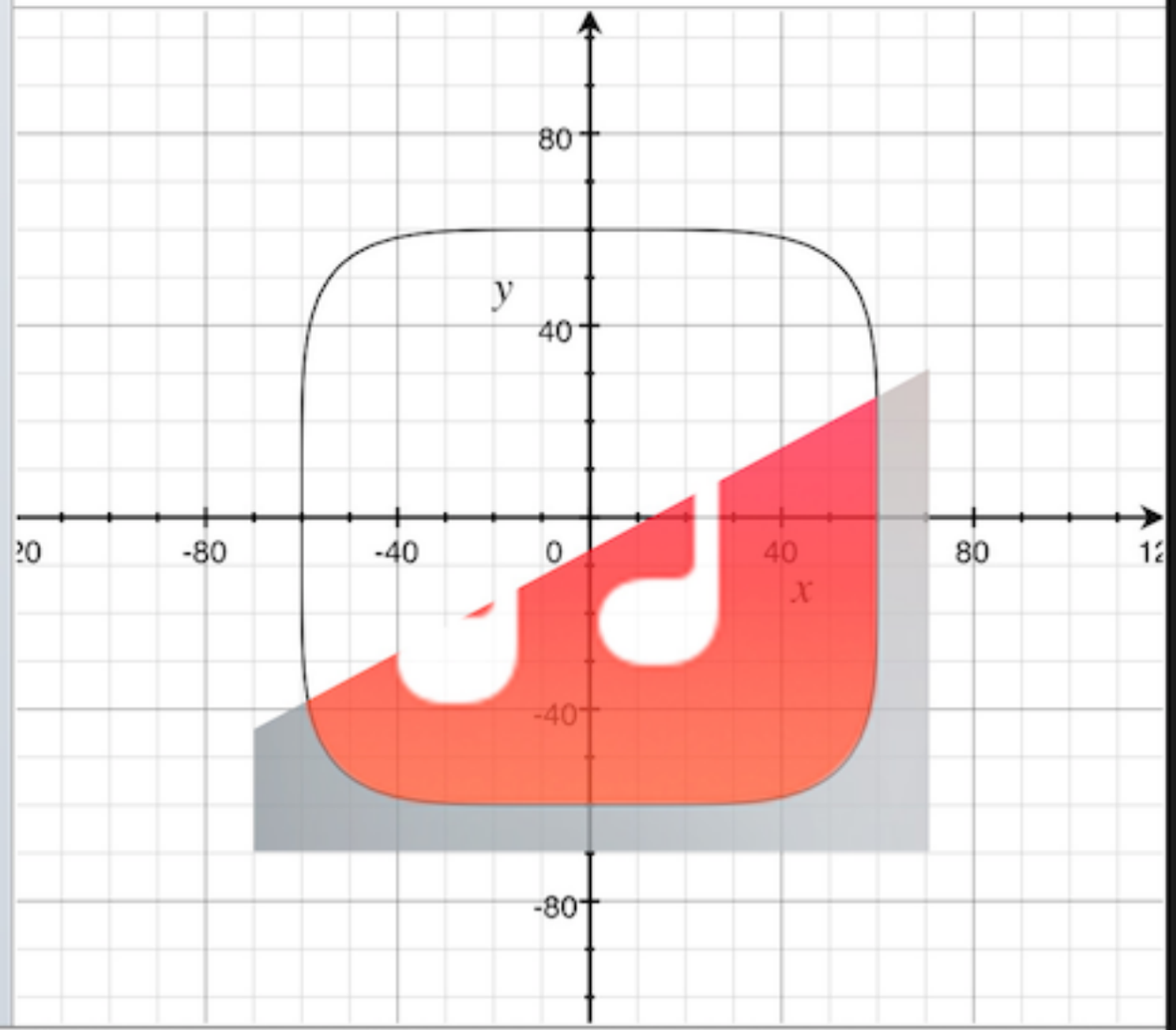
Depth

Icon radius: 12pt

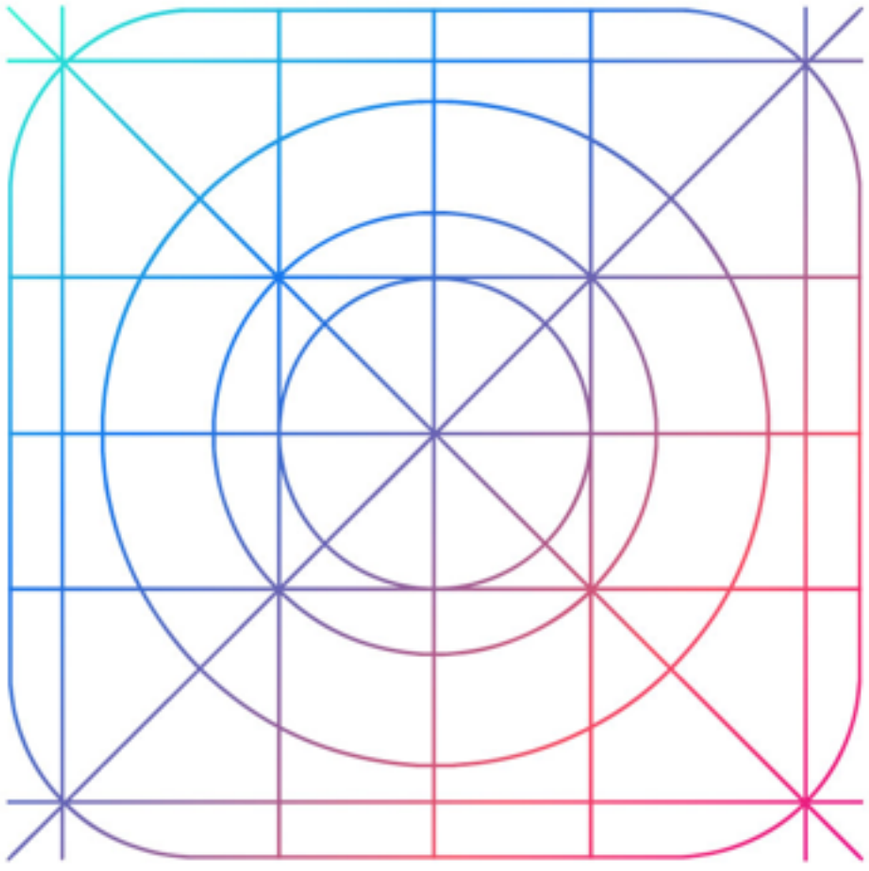
Superellipse

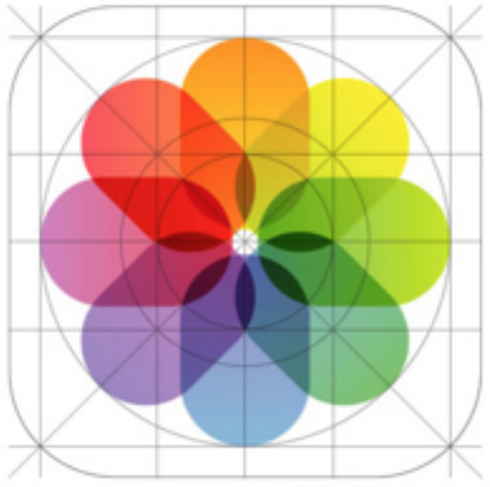
iOS 7 icon mask

$\left|\frac{x}{60}\right|^5+\left|\frac{y}{60}\right|^5=1$



@marcedwards





iOS 6

iOS 7


I'LL SPARE YOU
THE NUTS & BOLTS
MANUFACTURED
OR ASSIGNED

iOS 6


MANUFACTURED
OR ASSIGNED

iOS 7

iOS 6

iOS 7

iOS 6

iOS 7

iOS 6

iOS 7

Buttons are just text

Translucent Bars

Fullscreen Controllers are Good©®™

Update your apps:

Pay attention to translucent elements

Redesign icons and buttons

Prepare for borderless buttons

Beware of hard-coded UI locations

Use Dynamic Type

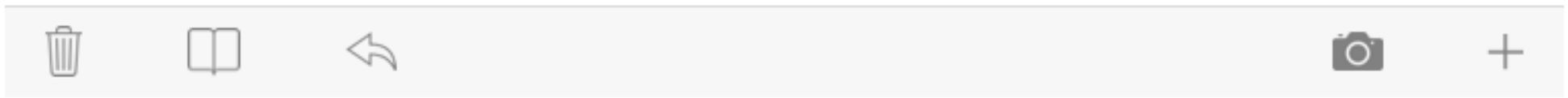
Pay attention to new gestures

Shadows, bezels, gradients

Use auto layout and (ahem!)
storyboards

Remove deprecated APIs

iOS 7 is a complete reboot for your app



Purple | Black | Orange | Green

Button

Button

(this is a UIProgressView)

Button

Button

Favorites | Recents | Contacts | More | Search

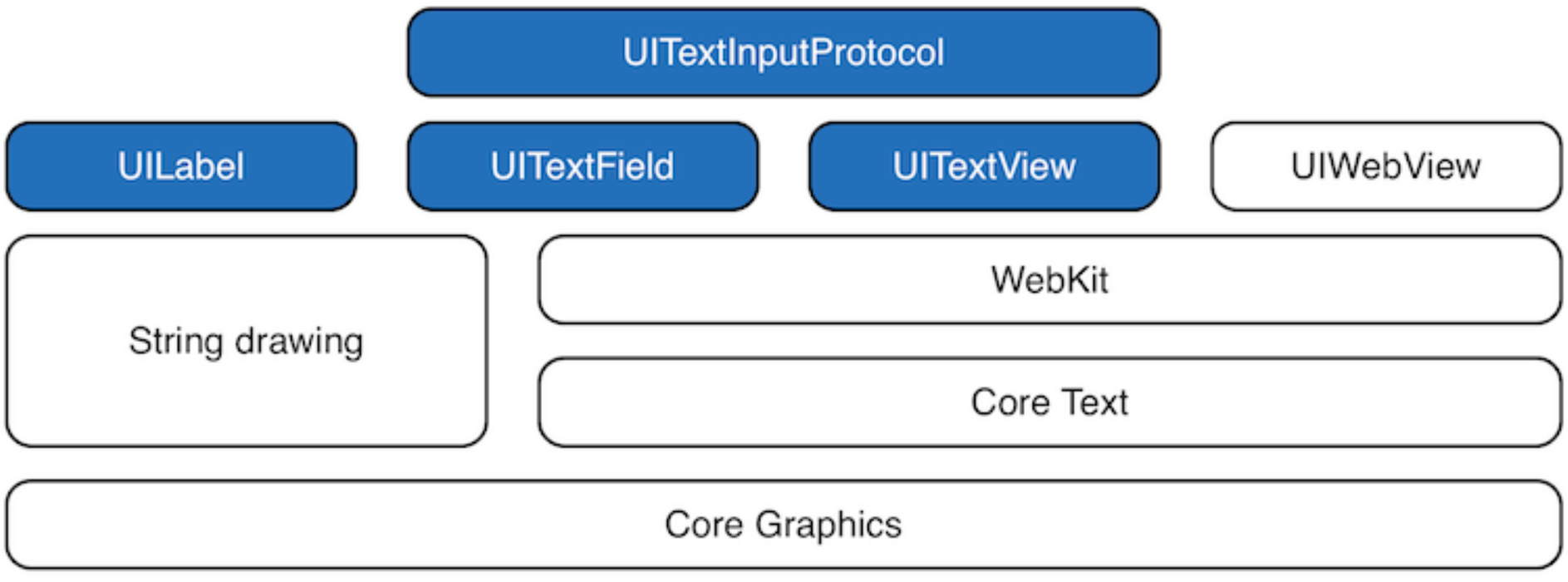
```
//
//  TRITintScreen.m
//  Presentation
//
//  Created by Adrian on 20/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRITintScreen.h"

@interface TRITintScreen ()

@property (weak, nonatomic) IBOutlet UITabBar *tabBar;
@property (weak, nonatomic) IBOutlet UITabBarItem *contactsItem;
@property (weak, nonatomic) IBOutlet UISegmentedControl *colorSwitch;

@end

@implementation TRITintScreen

+ (NSString *)xtype
{
    return @"tint";
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    self.tabBar.selectedItem = self.contactsItem;
}

- (void)performMainScreenAction
{
    self.colorSwitch.selectedSegmentIndex = 0;
}

- (IBAction)changeTint:(id)sender
{
    NSInteger index = [sender selectedSegmentIndex];
```

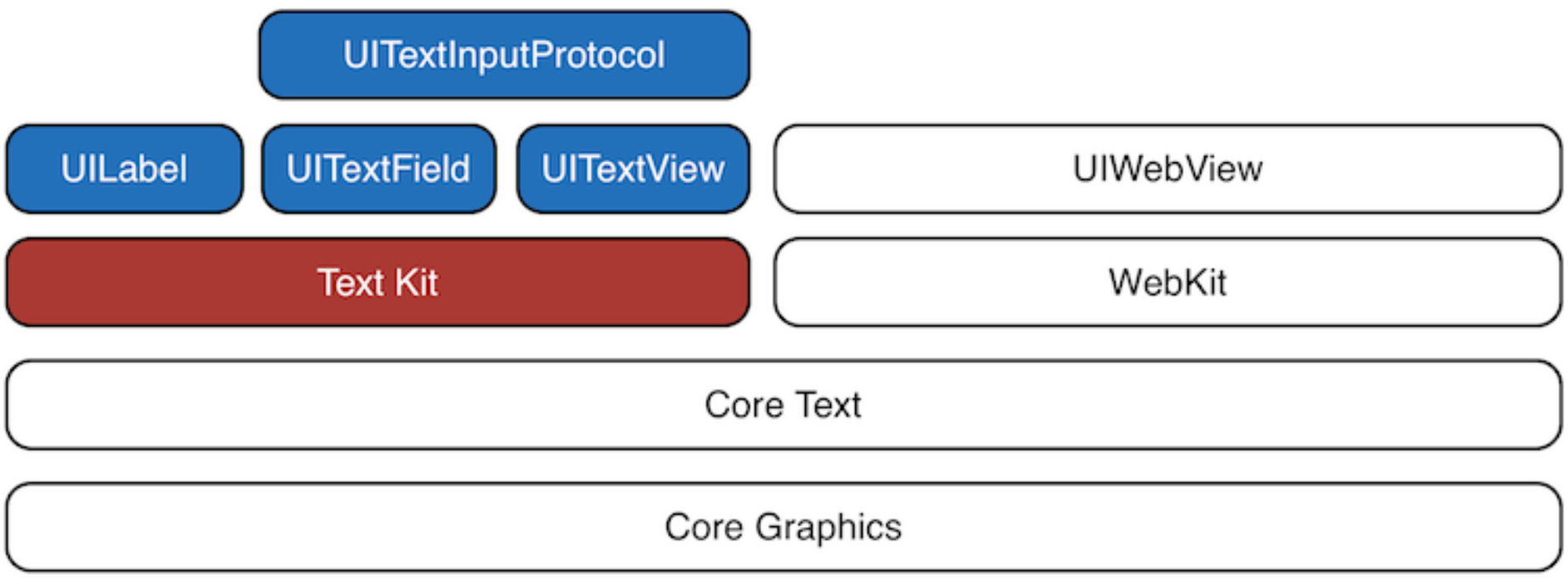
```
switch (index)
{
    case 0:
        self.view.tintColor = [UIColor purpleColor];
        break;

    case 1:
        self.view.tintColor = [UIColor blackColor];
        break;

    case 2:
        self.view.tintColor = [UIColor orangeColor];
        break;

    case 3:
        self.view.tintColor = [UIColor greenColor];
        break;

    default:
        break;
    }
}

@end
```

TextKit

Text processing for the masses

Probably the most important update in iOS 7


UITextInputProtocol
UILabel
UITextField
UITextView
UIWebView
String drawing
WebKit
Core Text
Core Graphics


UITextInputProtocol
UILabel
UITextField
UITextView
UIWebView
Text Kit
WebKit
Core Text
Core Graphics

Columns

Hyphenation

Kerning

Dynamic Type

Exclusion Paths

(aka "text wrapping an image")

```
textView.textContainer.exclusionPaths =
@[path];
```

(Latin: Confoederatio Helvetica, hence its abbreviation CH), is a federal parliamentary republic consisting of 26 cantons, with Bern as the seat of the federal authorities. The country is situated in Western and Central Europe,[note 4] where it is bordered by Germany to the north, France to the west, Italy to the south, and Austria and Liechtenstein to the east.

Switzerland is a landlocked country geographically divided between the Alps, the Swiss Plateau and the Jura, spanning an area of 41,285 km2 (15,940 sq mi). While the Alps occupy the greater part of the territory, the Swiss population of approximately 8 million people is concentrated mostly on the Plateau, where the largest cities are to be found. Among them are the two global cities and economic centres – Zürich and Geneva.

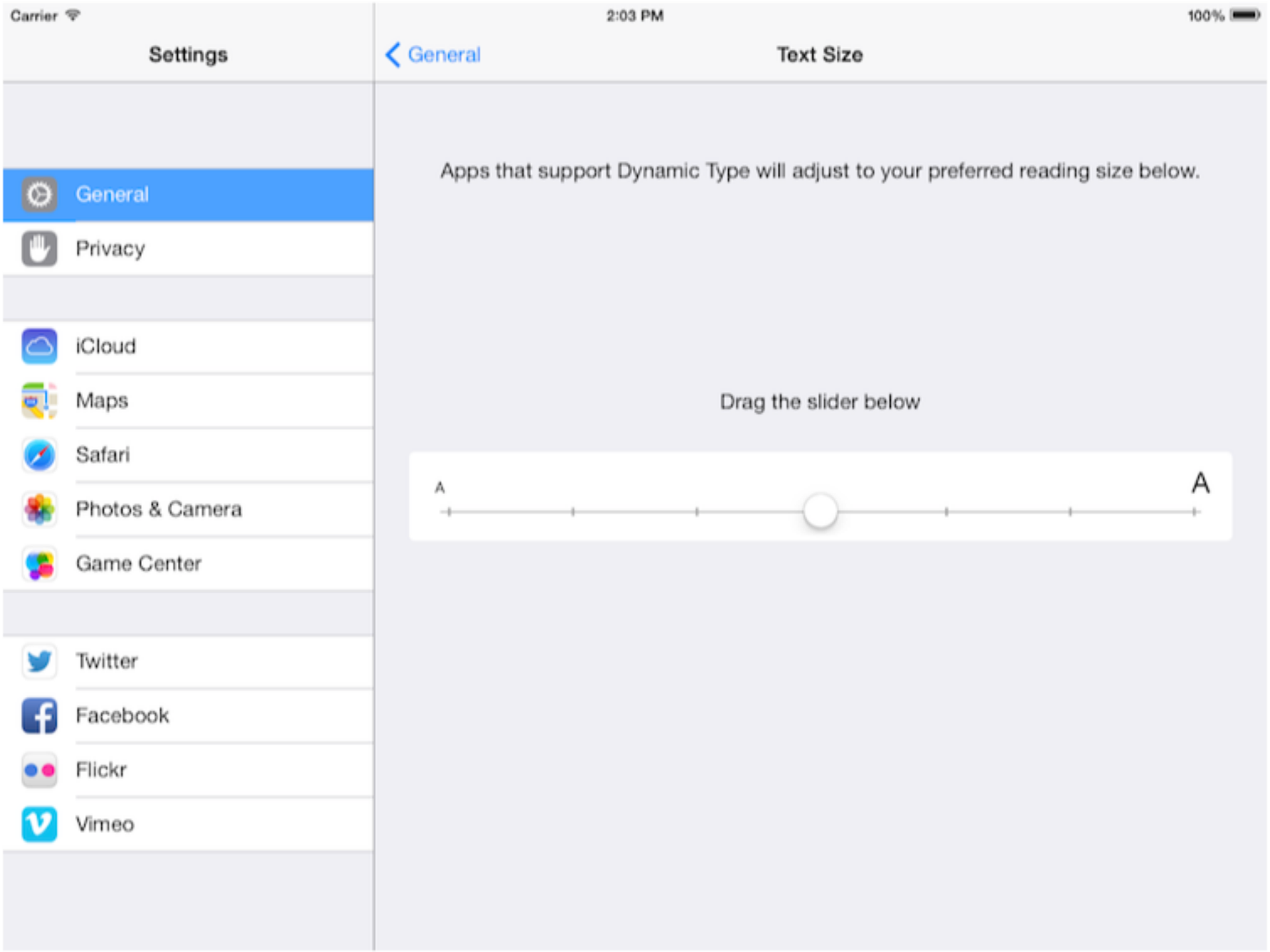
The Swiss Confederation has a long history of armed neutrality—it has not been in a state of war internationally since 1815—and did not join the United Nations until 2002. It pursues, however, an active foreign policy and is frequently involved in peace-building processes around the world.[8] Switzerland is also the birthplace of the Red Cross and home to a large number of international organizations, including the second largest UN office. On the European level, it is a founding member of the European Free Trade Association and is part of the Schengen Area – although it is notably not a member of the European Union, nor the European Economic Area.

Switzerland is one of the richest countries in the world by per capita gross domestic product, and has the highest wealth per adult (financial and non-financial assets) of any country in the world.[9][10] Zürich and Geneva have respectively been ranked as the

```
//
//  TRIInteractiveTextScreen.m
//  Presentation
//
//  Created by Adrian on 16/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//
// This code is adapted from
// https://github.com/objcio/issue-5-textkit
//
// The custom path is courtesy of
// https://github.com/Kjuly/UIBezierPath-Symbol
//

#import "TRIInteractiveTextScreen.h"
#import "TRIViews.h"
#import "TRIHelpers.h"

@interface TRIInteractiveTextScreen () <UITextViewDelegate>

@property (nonatomic) CGPoint offset;
@property (weak, nonatomic) IBOutlet UIView *crossView;
@property (weak, nonatomic) IBOutlet UITextView *textView;

@end


@implementation TRIInteractiveTextScreen

+ (NSString *)xtype
{
    return @"movetxt";
}

- (void)viewDidLoad
{
    [super viewDidLoad];
```

```
    // Enable hyphenation
    self.textView.layoutManager.hyphenationFactor = 1.0;
    [self updateExclusionPaths];
}

- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];
    [self updateExclusionPaths];
}

#pragma mark - Gesture recognizers

- (IBAction)pan:(UIPanGestureRecognizer *)pan
{
    if (pan.state == UIGestureRecognizerStateBegan)
    {
        self.offset = [pan locationInView:self.crossView];
    }
    else
    {
        CGPoint location = [pan locationInView:self.view];
        CGPoint center = self.crossView.center;

        CGFloat width = self.crossView.frame.size.width;
        center.x = location.x - self.offset.x + width / 2;
        center.y = location.y - self.offset.y + width / 2;
        self.crossView.center = center;

        [self updateExclusionPaths];
    }
}

#pragma mark - UIScrollViewDelegate methods

- (void)scrollViewDidScroll:(UIScrollView *)scrollView
{
    [self updateExclusionPaths];
}

#pragma mark - Private methods
```

```objc
- (void)updateExclusionPaths
{
    __weak typeof(self) weakSelf = self;
    dispatch_queue_priority_t prio = DISPATCH_QUEUE_PRIORITY_HIGH;
    dispatch_queue_t queue = dispatch_get_global_queue(prio, 0);
    dispatch_async(queue, ^{
        // Since text container does not know about the inset,
        // we must shift the frame to container coordinates
        CGRect frame = [weakSelf.textView convertRect:weakSelf.crossView.bounds
                                             fromView:weakSelf.crossView];

        frame.origin.x -= weakSelf.textView.textContainerInset.left;
        frame.origin.y -= weakSelf.textView.textContainerInset.top;

        // Set the exclusion path.
        UIBezierPath *path = [UIBezierPath customBezierPathOfPlusSymbolWithRect:frame
                                                                          scale:1.0];

        dispatch_async(dispatch_get_main_queue(), ^{
            weakSelf.textView.textContainer.exclusionPaths = @[path];
        });
    });
}

@end
```


Carrier
2:03 PM
100%
Settings
General
Text Size
General
Privacy
iCloud
Maps
Safari
Photos & Camera
Game Center
Twitter
Facebook
Flickr
Vimeo
Apps that support Dynamic Type will adjust to your preferred reading size below.
Drag the slider below
A
A

```
[UIFontDescriptor
preferredFontDescriptorWithTextStyle:
UIFontTextStyleBody]
```

UIContentSizeCategory…
…DidChangeNotification

```
//
//  TRISourceCodeScreen.m
//  Presentation
//
//  Created by Adrian on 13/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRISourceCodeScreen.h"
#import "TRIHelpers.h"


@interface TRISourceCodeScreen ()

@property (weak, nonatomic) IBOutlet UITextView *textView;
@property (nonatomic, strong) NSMutableAttributedString *sourceCode;

@end


@implementation TRISourceCodeScreen

+ (NSString *)xtype
{
    return @"source";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    UIBarButtonItem *closeButton = nil;
    UIBarButtonSystemItem type = UIBarButtonSystemItemDone;
```

```
//
//  TRISourceCodeScreen.m
//  Presentation
//
//  Created by Adrian on 13/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRISourceCodeScreen.h"
#import "TRIHelpers.h"


@interface TRISourceCodeScreen ()

@property (weak, nonatomic) IBOutlet UITextView *textView;
@property (nonatomic, strong) NSMutableAttributedString *sourceCode;

@end


@implementation TRISourceCodeScreen

+ (NSString *)xtype
{
    return @"source";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    UIBarButtonItem *closeButton = nil;
    UIBarButtonSystemItem type = UIBarButtonSystemItemDone;
    SEL sel = @selector(close:);
    closeButton = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:type
                                                                target:self
                                                                action:sel];
    self.navigationItem.rightBarButtonItem = closeButton;
```

```
    NSNotificationCenter *center = [NSNotificationCenter defaultCenter];
    [center addObserverForName:UIContentSizeCategoryDidChangeNotification
                        object:nil
                         queue:nil
                    usingBlock:^(NSNotification *notification) {
                        [self styleSourceCode];
                    }];
}

- (void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];

    if (self.className == nil)
    {
        self.className = NSStringFromClass([self class]);
    }
    else
    {
        NSString *title = [NSString stringWithFormat:@"%@.m", self.className];
        self.title = title;
    }

    self.sourceCode = [self loadSourceCode];
    self.textView.attributedText = self.sourceCode;
    [self styleSourceCode];
    [self.textView scrollRangeToVisible:NSMakeRange(0, 1)];
}

- (IBAction)close:(id)sender
{
    [self dismissViewControllerAnimated:YES
                             completion:nil];
}

#pragma mark - Private methods

- (NSMutableAttributedString *)loadSourceCode
{
    // Let's find out the path of the HTML file to show
    NSString *filename = [NSString stringWithFormat:@"html/%@.m", self.className];
```

```
NSString *path = [[NSBundle mainBundle] pathForResource:filename
                                                 ofType:@"html"];
NSMutableAttributedString *string = nil;

if (path)
{
    NSURL *url = [NSURL fileURLWithPath:path];

    // Now we're going to load that HTML on a mutable string
    NSDictionary *options = nil;
    options = @{
                NSDocumentTypeDocumentAttribute: NSHTMLTextDocumentType
                };
    NSError *error = nil;
    string = [[NSMutableAttributedString alloc] initWithFileURL:url
                                                        options:options
                                             documentAttributes:nil
                                                          error:&error];
}
    return string;
}

- (void)styleSourceCode
{
    NSMutableAttributedString *string = nil;
    NSAttributedString *text = self.textView.attributedText;
    string = [[NSMutableAttributedString alloc] initWithAttributedString:text];
    NSRange range = NSMakeRange(0, string.length - 1);

    NSString *style = UIFontTextStyleBody;
    UIFontDescriptor *descriptor = nil;
    descriptor = [UIFontDescriptor preferredFontDescriptorWithTextStyle:style];

    UIFont *newFont = [UIFont fontWithName:@"Menlo"
                                      size:descriptor.pointSize];
    [string addAttribute:NSFontAttributeName
                   value:newFont
                   range:range];

    self.textView.attributedText = string;
```

```
}

@end
```

NSHTMLTextDocumentType

UIKit Dynamics

The following programme contains moving objects, sensible viewers beware.

UIDynamicAnimator

UIPushBehavior

UIDynamicItemBehavior

UICollisionBehavior

```
//
//  TRIBehaviorScreen.m
//  Presentation
//
//  Created by Adrian on 20/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRIBehaviorScreen.h"
#import "TRIViews.h"

static long long const ARC4RANDOM_MAX = 0x100000000;


@interface TRIBehaviorScreen () <UICollisionBehaviorDelegate>

@property (strong, nonatomic) IBOutletCollection(TRICircleView) NSArray *circles;
@property (strong, nonatomic) UIDynamicAnimator *animator;

@end



@implementation TRIBehaviorScreen

+ (NSString *)xtype
{
    return @"behavior";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    self.animator = [[UIDynamicAnimator alloc] initWithReferenceView:self.view];

}
```

```objc
- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];
    for (TRICircleView *circle in self.circles)
    {
        circle.color = [self randomColor];

        UIPushBehavior *pusher = [[UIPushBehavior alloc] initWithItems:@[circle]
                                                                  mode:UIPushBehaviorModeInstantaneous];
        pusher.pushDirection = [self randomVector];
        pusher.active = YES;
        [self.animator addBehavior:pusher];

        UIDynamicItemBehavior *properties = nil;
        properties = [[UIDynamicItemBehavior alloc] initWithItems:@[circle]];
        properties.allowsRotation = NO;
        properties.elasticity = 1.0;
        properties.friction = 0.0;
        properties.resistance = 0.0;
        [self.animator addBehavior:properties];
    }

    UICollisionBehavior *collider = [[UICollisionBehavior alloc] initWithItems:self.circles];
    collider.collisionDelegate = self;
    collider.collisionMode = UICollisionBehaviorModeEverything;
    collider.translatesReferenceBoundsIntoBoundary = YES;
    [self.animator addBehavior:collider];
}

#pragma mark - Private methods

- (UIColor *)randomColor
{
    CGFloat red =  (CGFloat)arc4random() / (CGFloat)ARC4RANDOM_MAX;
    CGFloat blue = (CGFloat)arc4random() / (CGFloat)ARC4RANDOM_MAX;
    CGFloat green = (CGFloat)arc4random() / (CGFloat)ARC4RANDOM_MAX;
    return [UIColor colorWithRed:red green:green blue:blue alpha:1.0];
}

- (CGVector)randomVector
{
```

```
    CGFloat dx =  (CGFloat)arc4random() / (CGFloat)ARC4RANDOM_MAX;
    CGFloat dy = (CGFloat)arc4random() / (CGFloat)ARC4RANDOM_MAX;
    NSLog(@"vector: %1.2f, %1.2f", dx, dy);
    return CGVectorMake(dx, dy);
}

@end
```

NSURLSession

Remember ASIHTTPRequest and AFNetworking?
Yeah.

General | Capabilities | Info | Build Settings | Build Phases | Build Rules

PROJECT
Presentation

TARGETS
Presentation
Add Target...

▶ iCloud — OFF

▶ Game Center — OFF

▶ Passbook — OFF

▶ In-App Purchase — OFF

▶ Maps — OFF

▼ Background Modes — ON

Modes:
- [ ] Audio and AirPlay
- [ ] Location updates
- [ ] Voice over IP
- [ ] Newsstand downloads
- [ ] External accessory communication
- [ ] Uses Bluetooth LE accessories
- [ ] Acts as a Bluetooth LE accessory
- [x] Background fetch
- [ ] Remote notifications

Steps: ✓ Add the "Required Background Modes" key to your info plist file

▶ Keychain Sharing — OFF

▶ Inter-App Audio — OFF

▶ Data Protection — OFF

NSURLSession

NSURLSessionConfiguration

NSURLSessionDownloadTask

NSURLSessions are not NSOperationQueues

… but the idea is similar.

NSProgress

KVO Compliant

Parent/child relationships

Start download

Total size: 28 MB

Completed: 20 %

Suspend

Resume

Cancel

```
//
//  TRIURLSessionScreen.m
//  Presentation
//
//  Created by Adrian on 14/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRIURLSessionScreen.h"

static NSString *HUGE_WIKIPEDIA_IMAGE = @"http://upload.wikimedia.org/wikipedia/commons/9/9e/Paul_Gabri%C3%ABl_-
_Landscape_with_a_train_-_Google_Art_Project.jpg";


@interface TRIURLSessionScreen () <NSURLSessionDownloadDelegate>

@property (weak, nonatomic) IBOutlet UIProgressView *progressView;
@property (weak, nonatomic) IBOutlet UILabel *completedLabel;
@property (weak, nonatomic) IBOutlet UILabel *totalLabel;
@property (nonatomic) int64_t total;
@property (nonatomic, strong) NSURLSessionDownloadTask *task;
@property (nonatomic, strong) NSProgress *progress;
@property (nonatomic, strong) NSURLSession *session;

@end

@implementation TRIURLSessionScreen

+ (NSString *)xtype
{
    return @"download";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    self.total = 0;
    self.progressView.progress = 0.0;
```

```
    self.totalLabel.text = @"";
    self.completedLabel.text = @"";
}

- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];
    [self launchDownload:nil];
}

- (void)viewWillDisappear:(BOOL)animated
{
    [super viewWillDisappear:animated];
    [self.session invalidateAndCancel];
}

- (void)performMainScreenAction
{
    [self.session invalidateAndCancel];
    [self launchDownload:nil];
}

#pragma mark - IBAction methods

- (IBAction)launchDownload:(id)sender
{
    // Session configuration
    NSString *name = @"com.trifork.BackgroundDownloadSession";
    NSURLSessionConfiguration *conf = nil;
    conf = [NSURLSessionConfiguration backgroundSessionConfiguration:name];
    self.session = [NSURLSession sessionWithConfiguration:conf
                                                 delegate:self
                                            delegateQueue:nil];

    // Download task
    NSURL *url = [NSURL URLWithString:HUGE_WIKIPEDIA_IMAGE];
    NSURLRequest *request = [NSURLRequest requestWithURL:url];
    self.task = [self.session downloadTaskWithRequest:request];
    [self.task resume];

    // Progress object
```

```
    self.progress = [NSProgress progressWithTotalUnitCount:100];
    [self.progress addObserver:self
                    forKeyPath:@"completedUnitCount"
                       options:0
                       context:NULL];

    // Prepare the UI
    self.total = 0;
    self.progressView.progress = 0.0;
    self.totalLabel.text = @"";
    self.completedLabel.text = @"";
}

- (IBAction)cancel:(id)sender
{
    [self.session invalidateAndCancel];
}

- (IBAction)resume:(id)sender
{
    [self.task resume];
}

- (IBAction)suspend:(id)sender
{
    [self.task suspend];
}

#pragma mark - KVO

- (void)observeValueForKeyPath:(NSString *)keyPath
                      ofObject:(id)object
                        change:(NSDictionary *)change
                       context:(void *)context
{
    if (object == self.progress
        && [keyPath isEqualToString:@"completedUnitCount"])
    {
        dispatch_async(dispatch_get_main_queue(), ^{
            int64_t count = self.progress.completedUnitCount;
            self.progressView.progress = (float)count / 100.0f;
```

```
            self.totalLabel.text = [NSString stringWithFormat:@"%lld MB", self.total / 1024 / 1024];
            self.completedLabel.text = [NSString stringWithFormat:@"%lld %%", count];
        });
    }
}

#pragma mark - NSURLSessionDownloadDelegate methods

-        (void)URLSession:(NSURLSession *)session
            downloadTask:(NSURLSessionDownloadTask *)downloadTask
            didWriteData:(int64_t)bytesWritten
       totalBytesWritten:(int64_t)totalBytesWritten
totalBytesExpectedToWrite:(int64_t)totalBytesExpectedToWrite
{
    if (self.total == 0)
    {
        self.total = totalBytesExpectedToWrite;
    }
    self.progress.completedUnitCount = totalBytesWritten * 100 / totalBytesExpectedToWrite;
}


-        (void)URLSession:(NSURLSession *)session
            downloadTask:(NSURLSessionDownloadTask *)downloadTask
didFinishDownloadingToURL:(NSURL *)location
{
}



- (void)URLSession:(NSURLSession *)session
      downloadTask:(NSURLSessionDownloadTask *)downloadTask
 didResumeAtOffset:(int64_t)fileOffset
expectedTotalBytes:(int64_t)expectedTotalBytes
{
}


@end
```

Text-to-Speech

100% lip service

AVSpeechUtterance

AVSpeechSynthesizer

AVSpeechSynthesisVoice

Greetings from Switzerland!

Speak in English the text above

Minimum pitch

Default pitch

Maximum pitch

Available voices:

ar-SA

cs-CZ

da-DK

de-DE

el-GR

en-AU

en-GB

en-IE

```
//
//  TRISpeechScreen.m
//  Presentation
//
//  Created by Adrian on 13/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

@import AVFoundation;

#import "TRISpeechScreen.h"

static NSString *CELL_REUSE_IDENTIFIER = @"CELL_REUSE_IDENTIFIER";

@interface TRISpeechScreen () <UITableViewDelegate, UITableViewDataSource>

@property (weak, nonatomic) IBOutlet UITextView *inputText;
@property (weak, nonatomic) IBOutlet UITableView *tableView;
@property (nonatomic, strong) NSArray *speechVoices;
@property (nonatomic) float speechRate;
@property (nonatomic, copy) NSString *speechLanguage;

@end

@implementation TRISpeechScreen

+ (NSString *)xtype
{
    return @"speech";
}

#pragma mark - Speech methods


- (void)say:(NSString *)text withVoice:(AVSpeechSynthesisVoice *)voice
{
    AVSpeechUtterance *utterance = nil;
    utterance = [[AVSpeechUtterance alloc] initWithString:text];
    utterance.voice = voice;
```

```
    utterance.preUtteranceDelay = 0.3;
    utterance.rate = self.speechRate;

    AVSpeechSynthesizer *synth = [[AVSpeechSynthesizer alloc] init];
    [synth speakUtterance:utterance];
}



- (void)viewDidLoad
{
    [super viewDidLoad];

    // Get a sorted list of voices
    NSArray *voices = [AVSpeechSynthesisVoice speechVoices];
    NSComparisonResult (^cmp) (id, id) = ^NSComparisonResult (id obj1, id obj2) {
        return ([[obj1 language] compare:[obj2 language]]);
    };
    self.speechVoices = [voices sortedArrayUsingComparator:cmp];

    // Register a table view cell
    [self.tableView registerClass:[UITableViewCell class]
           forCellReuseIdentifier:CELL_REUSE_IDENTIFIER];

    self.speechRate = AVSpeechUtteranceDefaultSpeechRate;
    self.speechLanguage = @"en-EN";
}

- (void)performMainScreenAction
{
    [self speak:nil];
}

#pragma mark - IBActions

- (IBAction)speak:(id)sender
{
    NSString *text = self.inputText.text;
    AVSpeechSynthesisVoice *voice = [AVSpeechSynthesisVoice voiceWithLanguage:@"en-EN"];
```

```
    [self say:text withVoice:voice];
}

- (IBAction)speakAtMinimumRate:(id)sender
{
    self.speechRate = AVSpeechUtteranceMinimumSpeechRate;
    AVSpeechSynthesisVoice *voice = nil;
    voice = [AVSpeechSynthesisVoice voiceWithLanguage:self.speechLanguage];
    NSString *text = self.inputText.text;
    [self say:text withVoice:voice];
}

- (IBAction)speakAtDefaultRate:(id)sender
{
    self.speechRate = AVSpeechUtteranceDefaultSpeechRate;
    AVSpeechSynthesisVoice *voice = nil;
    voice = [AVSpeechSynthesisVoice voiceWithLanguage:self.speechLanguage];
    NSString *text = self.inputText.text;
    [self say:text withVoice:voice];
}

- (IBAction)speakAtMaximumRate:(id)sender
{
    self.speechRate = AVSpeechUtteranceMaximumSpeechRate;
    AVSpeechSynthesisVoice *voice = nil;
    voice = [AVSpeechSynthesisVoice voiceWithLanguage:self.speechLanguage];
    NSString *text = self.inputText.text;
    [self say:text withVoice:voice];
}

#pragma mark - Table view methods

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
 numberOfRowsInSection:(NSInteger)section
{
    return [self.speechVoices count];
```

```
}

-    (CGFloat)tableView:(UITableView *)tableView
heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
    return 44.0;
}

- (NSString *)tableView:(UITableView *)tableView
titleForHeaderInSection:(NSInteger)section
{
    return @"Available voices:";
}

- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    UITableViewCell *cell = nil;
    cell = [tableView dequeueReusableCellWithIdentifier:CELL_REUSE_IDENTIFIER
                                           forIndexPath:indexPath];
    cell.textLabel.font = [UIFont systemFontOfSize:26.0];
    AVSpeechSynthesisVoice *voice = self.speechVoices[indexPath.row];
    cell.textLabel.text = voice.language;
    return cell;
}

-       (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    NSString *text = self.inputText.text;
    AVSpeechSynthesisVoice *voice = self.speechVoices[indexPath.row];
    self.speechLanguage = voice.language;
    [self say:text withVoice:voice];
}

@end
```

Maps and Location

It's never been easier for the NSA to find you.

MKDistanceFormatter

MKDistanceFormatterUnits…
…Imperial
…Metric
…ImperialWithYards

(guess which one is the default?)

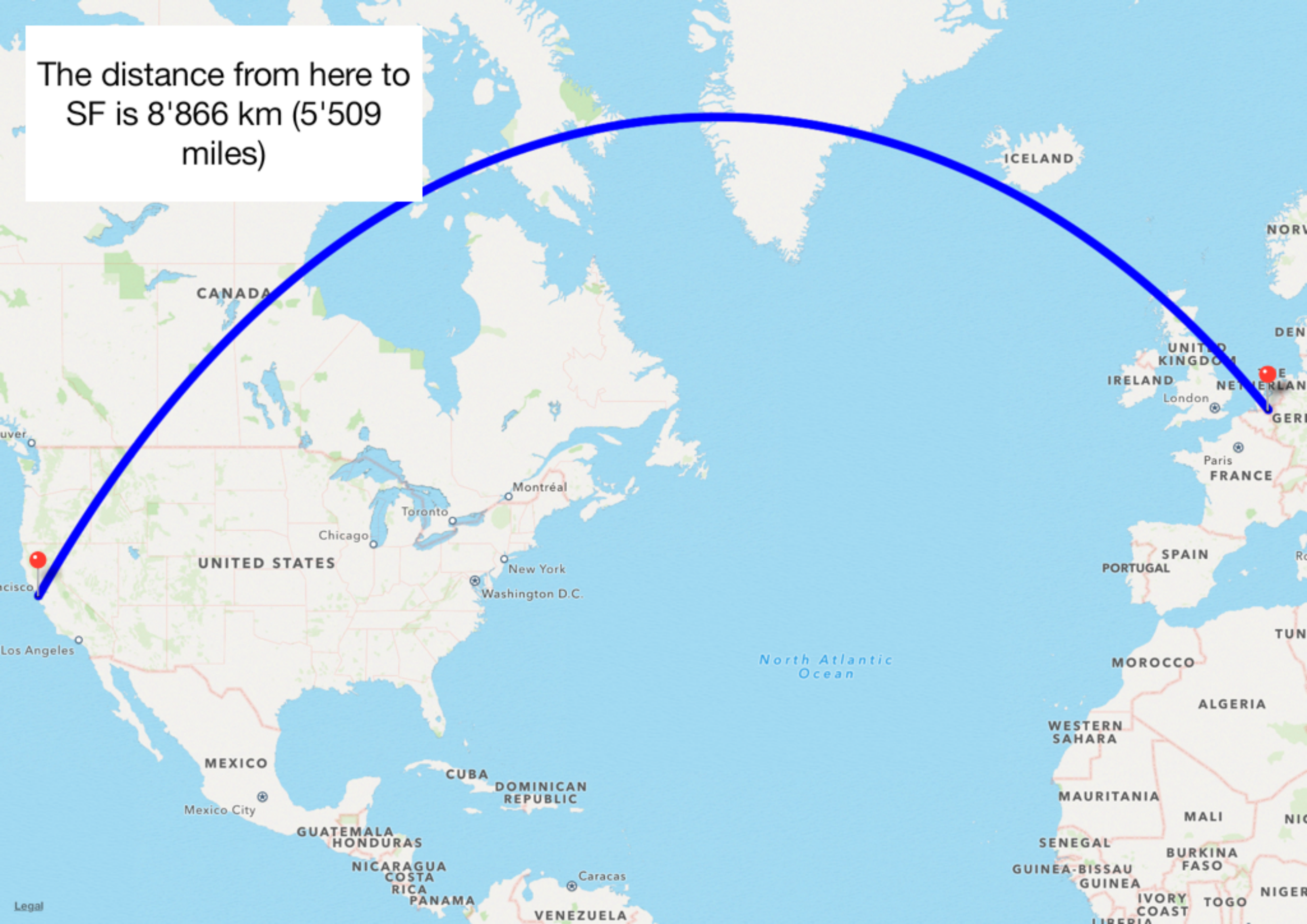
```
MKDistanceFormatterUnitsDefault
// depends on the locale!
```

MKGeodesicPolyline

MKPolylineRenderer

```
[map showAnnotations:array animated:YES]
```

The distance from here to SF is 8'866 km (5'509 miles)



```
//
//  TRIGeodesicScreen.m
//  Presentation
//
//  Created by Adrian on 13/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

@import CoreLocation;
@import MapKit;


#import "TRIGeodesicScreen.h"


static CLLocationDegrees SF_LATITUDE = 37.775056;
static CLLocationDegrees SF_LONGITUDE = -122.419321;
static NSString *REUSE_ID = @"REUSE_ID";


@interface TRIGeodesicScreen () <CLLocationManagerDelegate>

@property (weak, nonatomic) IBOutlet MKMapView *map;
@property (weak, nonatomic) IBOutlet UILabel *distanceLabel;
@property (nonatomic, strong) CLLocationManager *manager;
@property (nonatomic, strong) NSMutableArray *locations;

@end



@implementation TRIGeodesicScreen

+ (NSString *)xtype
{
    return @"geodesic";
}

- (void)viewDidLoad
```

```
{
    [super viewDidLoad];

    // We initialize the locations array with San Francisco
    self.locations = [NSMutableArray array];

    CLLocation *sf = nil;
    sf = [[CLLocation alloc] initWithLatitude:SF_LATITUDE
                                    longitude:SF_LONGITUDE];
    [self.locations addObject:sf];

    // We start the location manager for a short while
    self.manager = [[CLLocationManager alloc] init];
    self.manager.delegate = self;
    [self.manager startUpdatingLocation];
}

#pragma mark - CLLocationManagerDelegate methods

- (void)locationManager:(CLLocationManager *)manager
     didUpdateLocations:(NSArray *)locations
{
    if ([locations count] > 0)
    {
        // We get a second location and we draw a line
        // from here to SF.
        CLLocation *location = locations[0];
        [self.locations addObject:location];
        MKGeodesicPolyline *line = [self geodesicLineFrom:location];
        [self.map addOverlay:line];

        // This will zoom the map to show all the locations
        // passed as parameter
        [self.map showAnnotations:self.locations
                         animated:YES];

        // And we stop the location manager
        [self.manager stopUpdatingLocation];

        [self showDistanceToSF];
    }
```

```
}

- (MKGeodesicPolyline *)geodesicLineFrom:(CLLocation *)start
{
    // This is a c-array of MKMapPoint instances
    // required by the MKPolyline constructor below
    NSInteger count = 2;
    CLLocationCoordinate2D *coordinates = NULL;
    coordinates = malloc(sizeof(CLLocationCoordinate2D) * count);

    for (NSInteger index = 0; index < count; ++index)
    {
        CLLocation *loc = self.locations[index];
        coordinates[index] = loc.coordinate;
    }

    MKGeodesicPolyline *line = nil;
    line = [MKGeodesicPolyline polylineWithCoordinates:coordinates
                                                 count:count];
    free(coordinates);
    return line;
}

#pragma mark - MKMapViewDelegate methods

- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
            rendererForOverlay:(id<MKOverlay>)overlay
{
    if ([overlay isKindOfClass:[MKPolyline class]])
    {
        // The MKOverlayRenderer completely replaces the
        // old MKOverlayView class of yesteryear
        MKPolylineRenderer *renderer = nil;
        renderer = [[MKPolylineRenderer alloc] initWithOverlay:overlay];
        renderer.lineWidth = 5;
        renderer.fillColor = [UIColor blueColor];
        renderer.strokeColor = [UIColor blueColor];
        return renderer;
    }
    return nil;
```

```
}

- (MKAnnotationView *)mapView:(MKMapView *)mapView
           viewForAnnotation:(id <MKAnnotation>)annotation
{
    if (annotation == mapView.userLocation)
    {
        return nil;
    }

    MKAnnotationView *annotationView = nil;
    annotationView = [mapView dequeueReusableAnnotationViewWithIdentifier:REUSE_ID];
    if (annotationView == nil)
    {
        annotationView = [[MKPinAnnotationView alloc] initWithAnnotation:annotation
                                                         reuseIdentifier:REUSE_ID];
    }

    UIButton *rightButton = [UIButton buttonWithType:UIButtonTypeDetailDisclosure];
    annotationView.rightCalloutAccessoryView = rightButton;

    return annotationView;
}

#pragma mark - Private methods

- (void)showDistanceToSF
{
    // Show the distance using the new MKDistanceFormatter
    CLLocation *sf = self.locations[0];
    CLLocation *here = self.locations[1];
    CLLocationDistance distance = [sf distanceFromLocation:here];

    MKDistanceFormatter *formatter = [[MKDistanceFormatter alloc] init];
    formatter.units = MKDistanceFormatterUnitsMetric;
    NSString *km = [formatter stringFromDistance:distance];

    formatter.units = MKDistanceFormatterUnitsImperial;
    NSString *miles = [formatter stringFromDistance:distance];

    NSString *template = @"The distance from here to SF is %@ (%@)";
```

```
    NSString *message = [NSString stringWithFormat:template, km, miles];

    self.distanceLabel.hidden = NO;
    self.distanceLabel.text = message;

}

@end
```

iBeacon

Marketing people rejoice

TECH | 10/31/2013 @ 12:49PM | 8,056 views

How Apple's iBeacon Will Change Everything For Retail Analytics

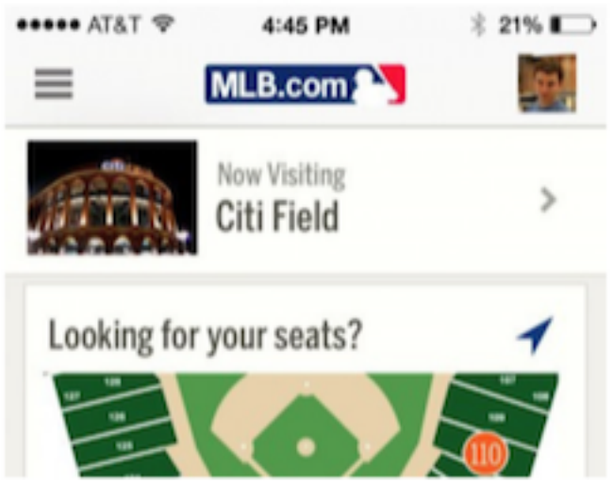
Natasha Baker, CenturyLink

The experience of shopping in a brick-and-mortar store hasn't changed much during the past 100 years: Consumers walk into a store, find the items they want, pay for the items and then lug their purchases out of the store in a shopping bag or cart.

But in-store experiences are on the cusp of a transformative shift, with Apple's iBeacon technology acting as a catalyst. Introduced in iOS 7, iBeacon is expected to change the way consumers interact with products and salespeople in stores, and could even alter the role that brick-and-mortar

http://www.youtube.com/watch?v=ITjsb22-EwQ

MLB demos using Apple's iBeacon technology to deliver personalised, interactive stadium experiences

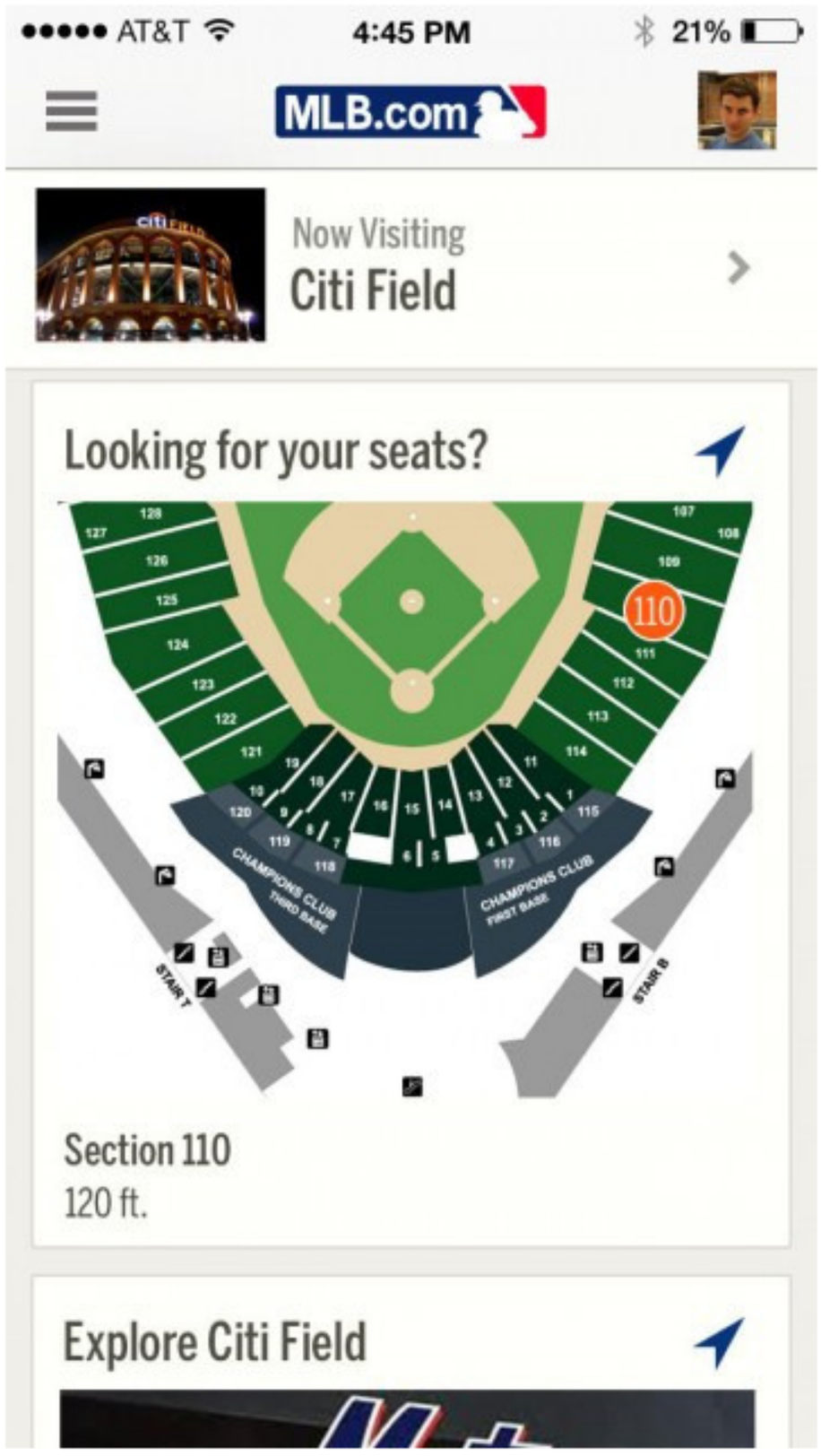


CATEGORY: IOS DEVICES

Written by:
BEN LOVEJOY
@benlovejoy
September 27, 2013 / 5:10 am

Apple Bluetooth LE Bluetooth Low Energy Citi Field Beacon iOS 7 iPhone Major League Baseball Mets Stadium

SHARE 8 COMMENTS


AT&T
4:45 PM
21%
MLB.com
Now Visiting
Citi Field
Looking for your seats?
128
127
126
125
124
123
122
121
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
120
119
118
117
116
115
114
113
112
111
110
109
108
107
CHAMPIONS CLUB
THIRD BASE
CHAMPIONS CLUB
FIRST BASE
STAIR 7
STAIR 8
Section 110
120 ft.
Explore Citi Field

CLBeaconRegion

CLBeacon

CLBeacon.proximityUUID

CLBeacon.major

CLBeacon.minor

Beacon found:	Yes
Proximity UUID:	E2C56DB5-DFFB-48D2-B060-D0F…
Major:	0
Minor:	0
Accuracy:	5.63
Distance:	Far
RSSI:	-75

```
//
//  TRIBeaconScreen.m
//  Presentation
//
//  Created by Adrian on 16/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

@import CoreLocation;

#import "TRIBeaconScreen.h"


static NSString *RASPBERRY_PI_UUID = @"E2C56DB5-DFFB-48D2-B060-D0F5A71096E0";


@interface TRIBeaconScreen () <CLLocationManagerDelegate>
@property (nonatomic, strong) CLLocationManager *manager;
@property (nonatomic, strong) CLBeaconRegion *region;

@property (weak, nonatomic) IBOutlet UILabel *foundLabel;
@property (weak, nonatomic) IBOutlet UILabel *uuidLabel;
@property (weak, nonatomic) IBOutlet UILabel *majorLabel;
@property (weak, nonatomic) IBOutlet UILabel *minorLabel;
@property (weak, nonatomic) IBOutlet UILabel *accuracyLabel;
@property (weak, nonatomic) IBOutlet UILabel *distanceLabel;
@property (weak, nonatomic) IBOutlet UILabel *rssiLabel;
@property (weak, nonatomic) IBOutlet UIActivityIndicatorView *spinningWheel;

@end

@implementation TRIBeaconScreen

+ (NSString *)xtype
{
    return @"ibeacon";
}
```

```objc
- (void)viewDidLoad
{
    [super viewDidLoad];

    self.manager = [[CLLocationManager alloc] init];
    self.manager.delegate = self;

    NSUUID *uuid = [[NSUUID alloc] initWithUUIDString:RASPBERRY_PI_UUID];
    self.region = [[CLBeaconRegion alloc] initWithProximityUUID:uuid
                                                     identifier:@"com.trifork.raspberrypi"];
    [self.manager startMonitoringForRegion:self.region];
}

#pragma mark - CLLocationManagerDelegate methods

- (void)locationManager:(CLLocationManager *)manager
         didEnterRegion:(CLRegion *)region
{
    [self.manager startRangingBeaconsInRegion:self.region];
}
```

```objc
-(void)locationManager:(CLLocationManager *)manager
         didExitRegion:(CLRegion *)region
{
    [self.manager stopRangingBeaconsInRegion:self.region];
    self.foundLabel.text = @"No";
}

-(void)locationManager:(CLLocationManager *)manager
       didRangeBeacons:(NSArray *)beacons
              inRegion:(CLBeaconRegion *)region
{
    // "firstObject" is another iOS 7 goodie!
    CLBeacon *beacon = [beacons firstObject];

    [self.spinningWheel stopAnimating];
    self.foundLabel.text = @"Yes";
    self.uuidLabel.text = beacon.proximityUUID.UUIDString;
    self.majorLabel.text = [beacon.major description];
    self.minorLabel.text = [beacon.minor description];
    self.accuracyLabel.text = [NSString stringWithFormat:@"%1.2f", beacon.accuracy];
```

```
    if (beacon.proximity == CLProximityUnknown)
    {
        self.distanceLabel.text = @"Unknown";
    }
    else if (beacon.proximity == CLProximityImmediate)
    {
        self.distanceLabel.text = @"Immediate";
    }
    else if (beacon.proximity == CLProximityNear)
    {
        self.distanceLabel.text = @"Near";
    }
    else if (beacon.proximity == CLProximityFar)
    {
        self.distanceLabel.text = @"Far";
    }
    self.rssiLabel.text = [NSString stringWithFormat:@"%li", (long)beacon.rssi];
}

@end
```

JavaScriptCore

It would have been faster if they had used Node.js instead.

JSVirtualMachine

JSContext

Collection of objects

Strings, numbers, blocks, etc

JSValue

[context evaluateScript:js]

```
var a = 10;
var factory = function (input) {
    return function () {
        return input + a;
    };
};
var fun = factory(50);
console.log(fun());
```

Execute

Clear

```
//
//  TRIJavaScriptScreen.m
//  Presentation
//
//  Created by Adrian on 17/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

@import JavaScriptCore;

#import "TRIJavaScriptScreen.h"


// This protocol states what is visible from the JS runtime

@protocol TRIJSExport <JSExport>

- (void)log:(NSString *)value;
- (void)clear;

@end


// Our console object redirects output to its associated text view

@interface TRIConsole : NSObject <TRIJSExport>

@property (nonatomic, weak) UITextView *textView;

- (instancetype)initWithTextView:(UITextView *)textView;

- (void)log:(NSString *)value;
- (void)clear;

@end



@implementation TRIConsole
```

```
- (instancetype)initWithTextView:(UITextView *)textView
{
    if (self = [super init])
    {
        self.textView = textView;
    }
    return self;
}

- (void)log:(NSString *)value
{
    NSMutableString *string = [NSMutableString stringWithString:self.textView.text];
    if ([string length] > 0)
    {
        [string appendString:@"\n"];
    }
    [string appendString:value];
    self.textView.text = string;

    NSUInteger location = [string length] - 2;
    NSRange range = NSMakeRange(location, 1);
    [self.textView scrollRangeToVisible:range];
}

- (void)clear
{
    self.textView.text = @"";
}

@end


// Finally, this is the screen itself

@interface TRIJavaScriptScreen ()

@property (weak, nonatomic) IBOutlet UITextView *editorTextView;
@property (weak, nonatomic) IBOutlet UITextView *consoleTextView;
@property (nonatomic, strong) JSContext *context;
@property (nonatomic, strong) TRIConsole *console;
```

```objc
@end


@implementation TRIJavaScriptScreen

+ (NSString *)xtype
{
    return @"javascript";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Initialize the virtual machine
    JSVirtualMachine *machine = [[JSVirtualMachine alloc] init];
    self.context = [[JSContext alloc] initWithVirtualMachine:machine];

    // The context can hold lots of objects
    self.context[@"copyright"] = @"Copyright Â© 2013 Trifork â€“ All Rights Reserved";
    self.context[@"author"] = @"Adrian Kosmaczewski";
    self.context[@"year"] = @2013;
    self.context[@"enabled"] = @YES;
    self.context[@"offices"] = @[ @"Ã…rhus", @"ZÃ¼rich", @"Amsterdam", @"Eindhoven" ];

    // Adding a console object to the context
    self.console = [[TRIConsole alloc] initWithTextView:self.consoleTextView];
    self.context[@"console"] = self.console;

    // We can also store blocks, and this one is recursive!
    // Leak-free recursive blocks, as explained by
    // http://jeremywsherman.com/blog/2013/02/27/leak-free-recursive-blocks/
    unsigned int (^__block __weak weakFibonacci) (unsigned int n);
    unsigned int (^fibonacci) (unsigned int n);
    weakFibonacci = fibonacci = ^unsigned int(unsigned int n) {
        if (n == 0)
        {
            return 0;
        }
        if (n == 1)
```

```
        {
            return 1;
        }
        return weakFibonacci(n - 1) + weakFibonacci(n - 2);
    };
    self.context[@"fibonacci"] = fibonacci;

    // What would be a JavaScript runtime without "alert()"?
    self.context[@"alert"] = ^ (NSString *text) {
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"JS Alert"
                                                        message:text
                                                       delegate:nil
                                              cancelButtonTitle:@"OK"
                                              otherButtonTitles:nil];
        [alert show];
    };
}

- (void)performMainScreenAction
{
    [self executeJavaScript:nil];
}

#pragma mark - IBAction methods

- (IBAction)executeJavaScript:(id)sender
{
    [self.editorTextView resignFirstResponder];
    NSString *js = self.editorTextView.text;
    JSValue *value = [self.context evaluateScript:js];
    NSLog(@"JSValue: %@", value);
}

- (IBAction)clearConsole:(id)sender
{
    [self.console clear];
}

@end
```

Smile Recognition

Siri can hear AND see you now.

```
detector = [CIDetector
  detectorOfType:CIDetectorTypeFace
    context:nil options:options];
```

```
[detector featuresInImage:image
options:@{ CIDetectorSmile: @YES }];
```


Not smiling :(

```
//
//  TRISmileScreen.m
//  Presentation
//
//  Created by Adrian on 19/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRISmileScreen.h"


@import AVFoundation;
@import CoreImage;



@interface TRISmileScreen () <AVCaptureVideoDataOutputSampleBufferDelegate>
@property (nonatomic, strong) AVCaptureSession *session;
@property (nonatomic, strong) AVCaptureVideoPreviewLayer *previewLayer;
@property (nonatomic, strong) AVCaptureVideoDataOutput *output;
@property (nonatomic, strong) CIDetector *detector;
@property (nonatomic) dispatch_queue_t queue;

@property (weak, nonatomic) IBOutlet UIView *previewView;
@property (weak, nonatomic) IBOutlet UITextView *textView;

@end

@implementation TRISmileScreen

+ (NSString *)xtype
{
    return @"smile";
}

- (void)viewDidLoad
{
    [super viewDidLoad];
```

```objc
// Adapted from
// http://www.renaudpradenc.com/?p=453
self.session = [[AVCaptureSession alloc] init];
AVCaptureDevice *device = [self cameraWithPosition:AVCaptureDevicePositionFront];

if(device != nil)
{
    // Set the input object
    NSError *error = nil;
    AVCaptureDeviceInput *input = nil;
    input = [AVCaptureDeviceInput deviceInputWithDevice:device
                                                  error:&error];

    if (error == nil)
    {
        if ([self.session canAddInput:input])
        {
            [self.session addInput:input];
        }

        // Create a detector object
        NSDictionary *options = @{
                                  CIDetectorTracking: @YES,
                                  CIDetectorAccuracy: CIDetectorAccuracyLow
                                  };
        self.detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                           context:nil
                                           options:options];

        // Create an output object
        self.output = [[AVCaptureVideoDataOutput alloc] init];

        self.queue = dispatch_queue_create("VideoDataOutputQueue", NULL);
        [self.output setSampleBufferDelegate:self
                                       queue:self.queue];

        if ([self.session canAddOutput:self.output])
        {
            [self.session addOutput:self.output];
        }
```

```
            // Create a preview layer
            self.previewLayer = nil;
            self.previewLayer = [[AVCaptureVideoPreviewLayer alloc] initWithSession:self.session];
            self.previewLayer.videoGravity = AVLayerVideoGravityResizeAspectFill;
            self.previewLayer.frame = self.previewView.bounds;
            [self.previewView.layer addSublayer:self.previewLayer];
            [self setPreviewOrientation];

            // Get those codes!
            [self.session startRunning];
        }
        else
        {
            NSLog(@"Error: %@", error);
        }
    }
}

- (void)didRotateFromInterfaceOrientation:(UIInterfaceOrientation)fromInterfaceOrientation
{
    [super didRotateFromInterfaceOrientation:fromInterfaceOrientation];
    [self setPreviewOrientation];
}

- (void)viewWillDisappear:(BOOL)animated
{
    [super viewWillDisappear:animated];
    [self.output setSampleBufferDelegate:nil queue:nil];
    self.output = nil;
    [self.session stopRunning];
    self.session = nil;
}

#pragma mark - AVCaptureVideoDataOutputSampleBufferDelegate methods

- (void)captureOutput:(AVCaptureOutput *)captureOutput
didOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer
       fromConnection:(AVCaptureConnection *)connection
{
    CVPixelBufferRef buffer = CMSampleBufferGetImageBuffer(sampleBuffer);
```

```
CIImage *image = [[CIImage alloc] initWithCVPixelBuffer:buffer];

NSArray *features = [self.detector featuresInImage:image
                                           options:@{ CIDetectorSmile: @YES }];

dispatch_async(dispatch_get_main_queue(), ^(void) {
    BOOL detected = NO;
    NSString *text = @"";
    for (CIFeature *feature in features)
    {
        if ([feature isKindOfClass:[CIFaceFeature class]])
        {
            CIFaceFeature *faceFeature = (CIFaceFeature *)feature;
            if (faceFeature.hasSmile)
            {
                detected = YES;
                text = @"Smiling! :)";
                break;
            }
        }
    }

    if (detected)
    {
        self.textView.text = text;
        self.textView.backgroundColor = [UIColor greenColor];
    }
    else
    {
        self.textView.text = @"Not smiling :(";
        self.textView.backgroundColor = [UIColor redColor];
    }
});
}

#pragma mark - Private methods

- (AVCaptureDevice *)cameraWithPosition:(AVCaptureDevicePosition)position
{
    NSArray *devices = [AVCaptureDevice devicesWithMediaType:AVMediaTypeVideo];
    for (AVCaptureDevice *device in devices)
```

```
    {
        if (device.position == position)
        {
            return device;
        }
    }
    return nil;
}

- (void)setPreviewOrientation
{
    AVCaptureVideoOrientation orientation = AVCaptureVideoOrientationPortrait;
    if (self.interfaceOrientation == UIInterfaceOrientationLandscapeLeft)
    {
        orientation = AVCaptureVideoOrientationLandscapeLeft;
    }
    else if (self.interfaceOrientation == UIInterfaceOrientationLandscapeRight)
    {
        orientation = AVCaptureVideoOrientationLandscapeRight;
    }
    else if (self.interfaceOrientation == UIInterfaceOrientationPortraitUpsideDown)
    {
        orientation = AVCaptureVideoOrientationPortraitUpsideDown;
    }
    self.previewLayer.connection.videoOrientation = orientation;
}

@end
```

Barcode Reading

Because QRCodes are the next big thing©®™

AVCaptureSession

AVCaptureDevice

AVCaptureDeviceInput

AVCaptureMetadataOutputObjectsDelegate

AVMetadataMachineReadableCodeObject

AVMetadataObjectType…
…UPCECode
…Code39Code
…Code39Mod43Code
…EAN13Code
…EAN8Code
…Code93Code
…Code128Code
…PDF417Code
…QRCode
…AztecCode

http://www.qrcode.com/

```
//
//  TRIQRCodeScreen.m
//  Presentation
//
//  Created by Adrian on 18/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRIQRCodeScreen.h"

@import AVFoundation;


@interface TRIQRCodeScreen () <AVCaptureMetadataOutputObjectsDelegate>

@property (nonatomic, strong) AVCaptureSession *session;
@property (nonatomic, strong) AVCaptureVideoPreviewLayer *previewLayer;

@property (weak, nonatomic) IBOutlet UIView *previewView;
@property (weak, nonatomic) IBOutlet UITextView *textView;

@end


@implementation TRIQRCodeScreen

+ (NSString *)xtype
{
    return @"qrcode";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Adapted from
    // http://www.renaudpradenc.com/?p=453
    self.session = [[AVCaptureSession alloc] init];
    AVCaptureDevice *device = [AVCaptureDevice defaultDeviceWithMediaType:AVMediaTypeVideo];
```

```
NSError *error = nil;

// Set the input object
AVCaptureDeviceInput *input = [AVCaptureDeviceInput deviceInputWithDevice:device
                                                                    error:&error];
if(input)
{
    [self.session addInput:input];

    // Set the output object
    AVCaptureMetadataOutput *output = [[AVCaptureMetadataOutput alloc] init];
    [self.session addOutput:output];
    dispatch_queue_t queue = dispatch_get_main_queue();
    [output setMetadataObjectsDelegate:self
                                 queue:queue];
    [output setMetadataObjectTypes:@[ AVMetadataObjectTypeQRCode ]];

    // Create a preview layer
    self.previewLayer = nil;
    self.previewLayer = [[AVCaptureVideoPreviewLayer alloc] initWithSession:self.session];
        self.previewLayer.videoGravity = AVLayerVideoGravityResizeAspectFill;
        self.previewLayer.frame = self.previewView.bounds;
        [self.previewView.layer addSublayer:self.previewLayer];

        // Get those codes!
        [self.session startRunning];
    }
    else
    {
        NSLog(@"Error: %@", error);
    }
}

- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];
    [self didRotateFromInterfaceOrientation:UIInterfaceOrientationPortrait];
}

- (void)didRotateFromInterfaceOrientation:(UIInterfaceOrientation)fromInterfaceOrientation
{
```

```
    AVCaptureVideoOrientation orientation = AVCaptureVideoOrientationPortrait;
    if (self.interfaceOrientation == UIInterfaceOrientationLandscapeLeft)
    {
        orientation = AVCaptureVideoOrientationLandscapeLeft;
    }
    else if (self.interfaceOrientation == UIInterfaceOrientationLandscapeRight)
    {
        orientation = AVCaptureVideoOrientationLandscapeRight;
    }
    else if (self.interfaceOrientation == UIInterfaceOrientationPortraitUpsideDown)
    {
        orientation = AVCaptureVideoOrientationPortraitUpsideDown;
    }
    self.previewLayer.connection.videoOrientation = orientation;
}

#pragma mark - AVCaptureMetadataOutputObjectsDelegate

-    (void)captureOutput:(AVCaptureOutput *)captureOutput
didOutputMetadataObjects:(NSArray *)metadataObjects
          fromConnection:(AVCaptureConnection *)connection
{
    for (AVMetadataObject *metadata in metadataObjects)
    {
        if ([metadata.type isEqualToString:AVMetadataObjectTypeQRCode])
        {
            NSString *code = [(AVMetadataMachineReadableCodeObject *)metadata stringValue];
            self.textView.text = code;
            break;
        }
    }

}

@end
```

```
// Behold!

[AVCaptureSession addOutput:]
// must be called before
[AVCaptureMetadataOutput
setMetadataObjectTypes:]
```

Web Development in iOS 7

Bad News Coming

"I can say without fear of mistake that this is the buggiest Safari version since 1.0."

Maximiliano Firtman

Full screen webapps problems

'online' and 'offline' events not firing

No more <input type="datetime">

No WebGL or IndexedDB yet

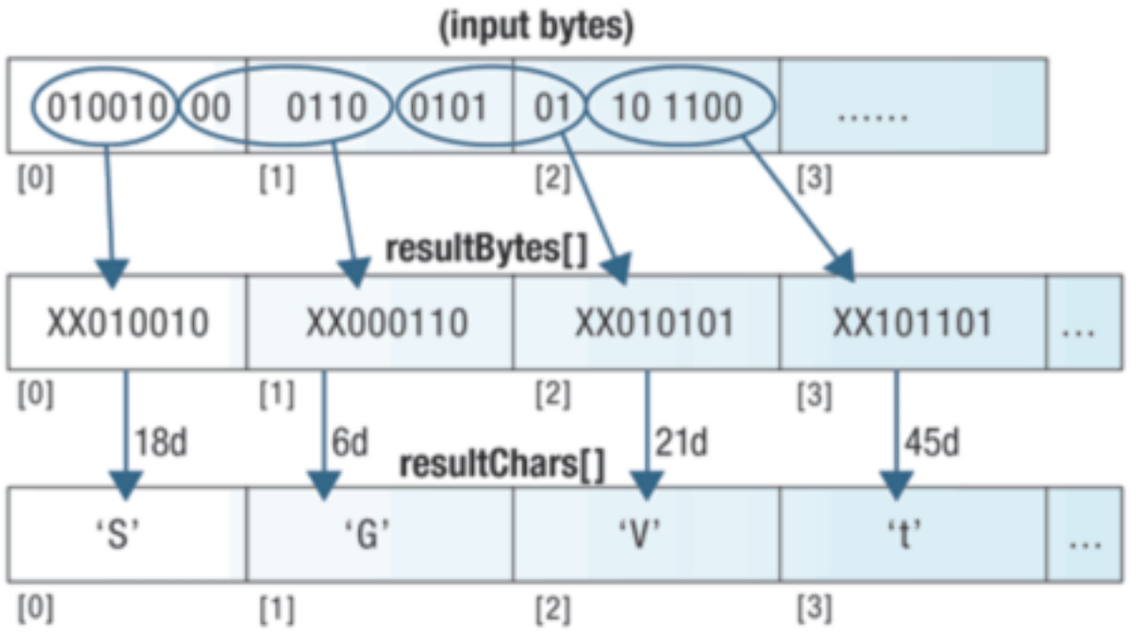
Serious WebSQL bugs

New 'Add to Home Screen' button
New UI gestures
New icon sizes
AirPlay API
CSS Grid Layout

More!

KISS principle applies here

```
[NSData base64EncodedStringWithOptions:]
```


(input bytes)
010010
00
0110
0101
01
10 1100
.....
[0]
[1]
[2]
[3]
resultBytes[]
XX010010
XX000110
XX010101
XX101101
...
[0]
[1]
[2]
[3]
18d
6d
21d
45d
resultChars[]
'S'
'G'
'V'
't'
...
[0]
[1]
[2]
[3]

```
//
//  TRIBase64Screen.m
//  Presentation
//
//  Created by Adrian on 18/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#import "TRIBase64Screen.h"

@interface TRIBase64Screen ()

@property (weak, nonatomic) IBOutlet UIWebView *webView;

@end


@implementation TRIBase64Screen

+ (NSString *)xtype
{
    return @"base64";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    UIImage *image = [UIImage imageNamed:@"base64"];
    NSData *data = UIImagePNGRepresentation(image);
    NSString *base64 = [data base64EncodedStringWithOptions:0];
    NSString *img = @"<p align=\"center\" style=\"margin: 50px\"><img src=\"%@\" alt=\"base64 Test\"></
p>";
    NSString *html = [NSString stringWithFormat:img, base64];
    [self.webView loadHTMLString:html
                         baseURL:nil];
}

@end
```

SSReadingList

Add Trifork.com to your Safari Reading List!

```
//
//  TRIReadingListScreen.m
//  Presentation
//
//  Created by Adrian on 19/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

@import SafariServices;


#import "TRIReadingListScreen.h"


@implementation TRIReadingListScreen

+ (NSString *)xtype
{
    return @"readinglist";
}

- (IBAction)addToReadingList:(id)sender
{
    NSURL *URL = [NSURL URLWithString:@"http://trifork.com"];
    [[SSReadingList defaultReadingList] addReadingListItemWithURL:URL
                                                            title:@"Trifork"
                                                      previewText:@"..."
                                                            error:nil];
}

@end
```

CMStepCounter

Start counting steps

Stop counting steps

Step counting not available in this device

```
//
//  TRICountingStepsScreen.m
//  Presentation
//
//  Created by Adrian on 19/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

@import CoreMotion;


#import "TRICountingStepsScreen.h"



@interface TRICountingStepsScreen ()

@property (weak, nonatomic) IBOutlet UILabel *stepsLabel;
@property (strong, nonatomic) CMStepCounter *stepCounter;
@property (nonatomic) NSOperationQueue *queue;

@end



@implementation TRICountingStepsScreen

+ (NSString *)xtype
{
    return @"steps";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    if ([CMStepCounter isStepCountingAvailable])
    {
        self.stepCounter = [[CMStepCounter alloc] init];
```

```
        self.queue = [NSOperationQueue mainQueue];
    }
    else
    {
        self.stepsLabel.text = @"Step counting not available in this device";
    }
}

- (IBAction)startCountingSteps:(id)sender
{
    CMStepUpdateHandler handler = ^(NSInteger numberOfSteps, NSDate *timestamp, NSError *error) {
        self.stepsLabel.text = [NSString stringWithFormat:@"Steps: %ld", (long)numberOfSteps];
    };
    self.stepsLabel.text = @"";
    [self.stepCounter startStepCountingUpdatesToQueue:self.queue
                                             updateOn:1
                                          withHandler:handler];
}

- (IBAction)stopCountingSteps:(id)sender
{
    [self.stepCounter stopStepCountingUpdates];
}

@end
```

No More MAC Address

(privacy concerns)

02:00:00:00:00:00

```
//
//  TRIMACAddressScreen.m
//  Presentation
//
//  Created by Adrian on 19/11/13.
//  Copyright (c) 2013 Trifork GmbH. All rights reserved.
//

#include <sys/socket.h>
#include <sys/sysctl.h>
#include <net/if.h>
#include <net/if_dl.h>

#import "TRIMACAddressScreen.h"


@interface TRIMACAddressScreen ()
@property (weak, nonatomic) IBOutlet UILabel *label;

@end



@implementation TRIMACAddressScreen

+ (NSString *)xtype
{
    return @"macaddress";
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    self.label.text = [self macAddress];
}

#pragma mark - Very private parts
```

```objc
- (NSString *)macAddress
{
    // Courtesy of
    // http://stackoverflow.com/q/14827365/133764
    int                 mgmtInfoBase[6];
    char                *msgBuffer = NULL;
    NSString            *errorFlag = NULL;
    size_t              length;

    // Setup the management Information Base (mib)
    mgmtInfoBase[0] = CTL_NET;        // Request network subsystem
    mgmtInfoBase[1] = AF_ROUTE;       // Routing table info
    mgmtInfoBase[2] = 0;
    mgmtInfoBase[3] = AF_LINK;        // Request link layer information
    mgmtInfoBase[4] = NET_RT_IFLIST;  // Request all configured interfaces

    // With all configured interfaces requested, get handle index
    if ((mgmtInfoBase[5] = if_nametoindex("en0")) == 0)
        errorFlag = @"if_nametoindex failure";
    // Get the size of the data available (store in len)
    else if (sysctl(mgmtInfoBase, 6, NULL, &length, NULL, 0) < 0)
        errorFlag = @"sysctl mgmtInfoBase failure";
    // Alloc memory based on above call
    else if ((msgBuffer = malloc(length)) == NULL)
        errorFlag = @"buffer allocation failure";
    // Get system information, store in buffer
    else if (sysctl(mgmtInfoBase, 6, msgBuffer, &length, NULL, 0) < 0)
    {
        free(msgBuffer);
        errorFlag = @"sysctl msgBuffer failure";
    }
    else
    {
        // Map msgbuffer to interface message structure
        struct if_msghdr *interfaceMsgStruct = (struct if_msghdr *) msgBuffer;

        // Map to link-level socket structure
        struct sockaddr_dl *socketStruct = (struct sockaddr_dl *) (interfaceMsgStruct + 1);

        // Copy link layer address data in socket structure to an array
```

```
        unsigned char macAddress[6];
        memcpy(&macAddress, socketStruct->sdl_data + socketStruct->sdl_nlen, 6);

        // Read from char array into a string object, into traditional MAC address format
        NSString *macAddressString = [NSString stringWithFormat:@"%02X:%02X:%02X:%02X:%02X:%02X",
                                      macAddress[0], macAddress[1], macAddress[2],
                                      macAddress[3], macAddress[4], macAddress[5]];

        // Release the buffer memory
        free(msgBuffer);

        return macAddressString;
    }

    return errorFlag;
}

@end
```

… and soooo much more…

Thanks for attending!

Get the PDF of this presentation
while it's hot.

Credits:

Github
objc.io #5
Wikipedia
NSHipster
DevFright.com
Stack Overflow
Typographica.org
Big Nerd Ranch Blog
Renaud Pradenc Blog
Maximiliano Firtman Blog
CapTech Consulting Blog

iOS Training in the Netherlands in January!

Contact us for more details.

ako@trifork.com

@akotrifork

Questions?

TRIFORK.

...think software