

«MyView» Objective-C handmade implementation

Fabrice.Kordon@lip6.fr



As an introduction



Programmatic implementation

- Objective-C
- Explicit handling of coordinates
 - ▶ The «increment method»
- No «model» here
 - ▶ Far too simple



Demo on a 4" screen



Demo on a 5.8" screen

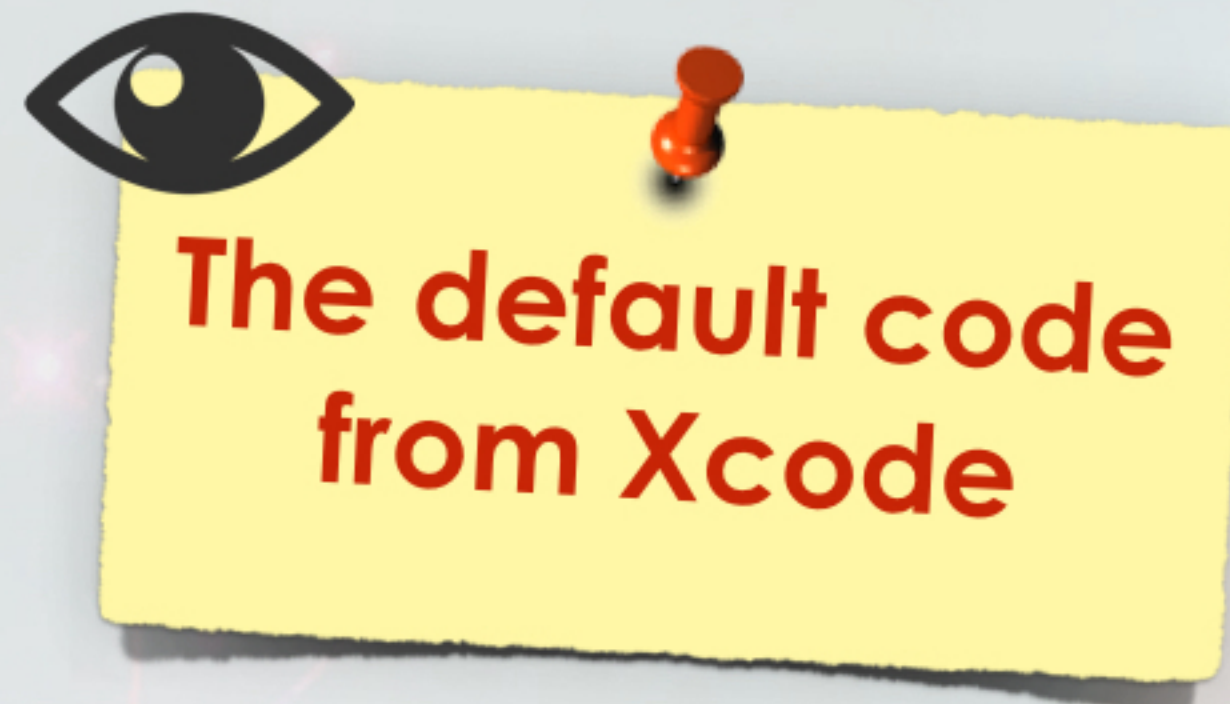


Demo on a 9.7" screen



ViewController.h

```
//  
// ViewController.h  
// MyView-objc  
//  
// Created by Fabrice Kordon on 27/09/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
#import <UIKit/UIKit.h>  
  
@interface ViewController : UIViewController  
  
@end
```



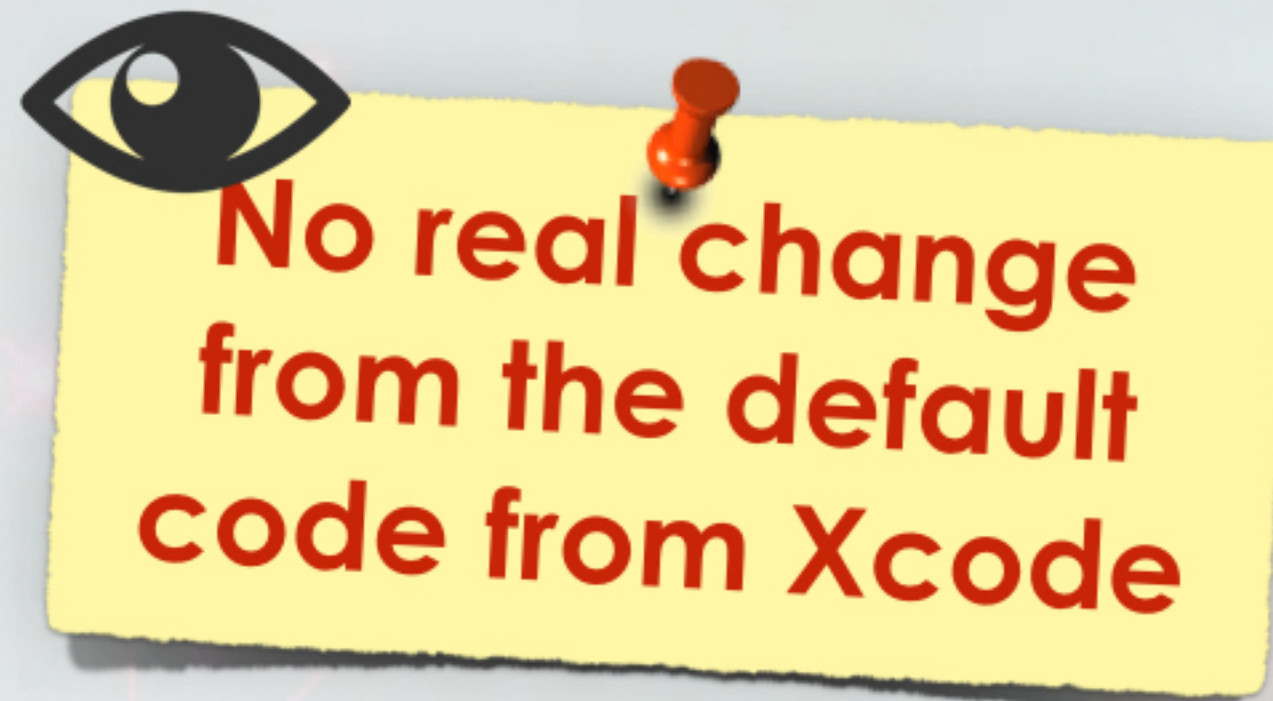
ViewController.m

7

```
//  
// ViewController.m  
// MavueObjc  
//  
// Created by Fabrice Kordon on 11/10/2016.  
// Copyright © 2016 Fabrice Kordon. All rights reserved.  
//  
  
#import "ViewController.h"  
#import "MyView.h"  
  
@interface ViewController ()  
  
@end  
  
@implementation ViewController  
  
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
    UIScreen *ecran = [UIScreen mainScreen];  
    CGRect rect = [ecran bounds];  
    MyView *v = [[MyView alloc] initWithFrame:rect];  
    [self setView:v];  
    [v release]; // because view is qualified as «retain»  
}  
  
@end
```

MyView.h

```
//  
// MyView.h  
// MyView-objc  
//  
// Created by Fabrice Kordon on 27/09/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
#import <UIKit/UIKit.h>  
  
@interface MyView : UIView  
  
@end
```



MyView.m

```
#import "MyView.h"  
  
@implementation MyView  
  
UILabel *deviceData;  
UILabel *model;  
UILabel *orientation;  
UIDevice *myDevice;  
int top, incr;
```

MyView.m

```
- (id) initWithFrame:(CGRect)frame { // Let's init the static elements
    if ([super initWithFrame:frame]) {
        [self setBackgroundColor:[UIColor whiteColor]];
        myDevice = [UIDevice currentDevice];
        UIScreen *screen = [UIScreen mainScreen];
        deviceData = [[UILabel alloc] init];
        [deviceData setTextAlignment:NSTextAlignmentCenter];
        NSLog(@"[screen scale] = %.1f", [screen scale]);
        if ([myDevice userInterfaceIdiom] == UIUserInterfaceIdiomPhone &&
            [screen scale] == 3.0) {
            [deviceData setText:[NSString
                stringWithFormat:@"Large iPhone (%@, %@)",
                [myDevice systemName], [myDevice systemVersion]]];

            incr = 50;
        } else if ([myDevice userInterfaceIdiom] == UIUserInterfaceIdiomPhone){
            [deviceData setText:[NSString stringWithFormat:@"iPhone/iPod (%@, %@)",
                [myDevice systemName], [myDevice systemVersion]]];

            incr = 40;
        } else {
            [deviceData setText:[NSString stringWithFormat:@"iPad (%@, %@)",
                [myDevice systemName], [myDevice systemVersion]]];

            incr = 100;
        }
    }
}
```

MyView.m

```
// Let's deal with the view's elements
model = [[UILabel alloc] init];
[model setTextAlignment:NSTextAlignmentCenter];
[model setText:[myDevice model]];
orientation = [[UILabel alloc] init];
[orientation setTextAlignment:NSTextAlignmentCenter];
[self addSubview:deviceData];
[deviceData release];
[self addSubview:model];
[model release];
[self addSubview:orientation];
[orientation release];
}
return self;
}
```

MyView.m

```
- (void)drawRect:(CGRect)rect { // display the view's content
    CGFloat h = rect.size.height;
    CGFloat w = rect.size.width;

    if ([myDevice orientation] == UIDeviceOrientationPortrait ||
        [myDevice orientation] == UIDeviceOrientationPortraitUpsideDown) {
        [orientation setText:@"Portrait orientation"];
        top = 100;
    } else if ([myDevice orientation] == UIDeviceOrientationLandscapeLeft ||
               [myDevice orientation] == UIDeviceOrientationLandscapeRight) {
        [orientation setText:@"Landscape orientation"];
        top = 50;
    } else {
        [orientation setText:@"device is laid flat"];
    }
    [deviceData setFrame:CGRectMake(w / 2 - 150, top, 300, 21)];
    [model setFrame:CGRectMake(w / 2 - 150, top + incr, 300, 21)];
    [orientation setFrame:CGRectMake(w / 2 - 150, top + 2 * incr, 300, 21)];
}
```

MyView.m

```
// Let's draw some circles
[[UIColor colorWithRed:1.0 green:0.9 blue:0.0 alpha:1.0] setFill];
CGContextRef ctx = UIGraphicsGetCurrentContext();
CGContextAddArc(ctx, w / 2, h / 2 + 70, 70, 0, M_PI * 2, YES);
CGContextFillPath(ctx);
[[UIColor colorWithRed:1.0 green:0.8 blue:0.0 alpha:1.0] setFill];
CGContextAddArc(ctx, w / 2, h / 2 + 70, 60, 0, M_PI * 2, YES);
CGContextFillPath(ctx);
[[UIColor colorWithRed:1.0 green:0.7 blue:0.0 alpha:1.0] setFill];
CGContextAddArc(ctx, w / 2, h / 2 + 70, 50, 0, M_PI * 2, YES);
CGContextFillPath(ctx);
[[UIColor colorWithRed:1.0 green:0.6 blue:0.0 alpha:1.0] setFill];
CGContextAddArc(ctx, w / 2, h / 2 + 70, 40, 0, M_PI * 2, YES);
CGContextFillPath(ctx);
[[UIColor colorWithRed:1.0 green:0.5 blue:0.0 alpha:1.0] setFill];
CGContextAddArc(ctx, w / 2, h / 2 + 70, 30, 0, M_PI * 2, YES);
CGContextFillPath(ctx);
[[UIColor colorWithRed:1.0 green:0.3 blue:0.0 alpha:1.0] setFill];
CGContextAddArc(ctx, w / 2, h / 2 + 70, 20, 0, M_PI * 2, YES);
CGContextFillPath(ctx);
[[UIColor colorWithRed:1.0 green:0.0 blue:0.0 alpha:1.0] setFill];
CGContextAddArc(ctx, w / 2, h / 2 + 70, 10, 0, M_PI * 2, YES);
CGContextFillPath(ctx);
```

```
}
```

MyView.m

```
- (void)dealloc {  
    // Nothing to do because appropriately handled in  
    // the viewDidLoad() method  
    [super dealloc];  
}  
  
@end
```

As a conclusion...



Easy is'n't it?

- Do you really need StoryBoard?
 - ▶ A first way to deal with orientation



- You just have to think a bit

