

Example «uRotate»

Fabrice.Kordon@lip6.fr



As an introduction...

A simple application handling orientation

- Programmatic elaboration of the interface
- Show how it works by the example

uRotate

- Objective-C
- Swift
- «hide and seek» with some images
 - ▶ Use of a button too

Demo (4" screen)



Demo (5.8" screen)



Demo (9.7" screen)



Architecture



AppDelegate

- Do not touch (yet)



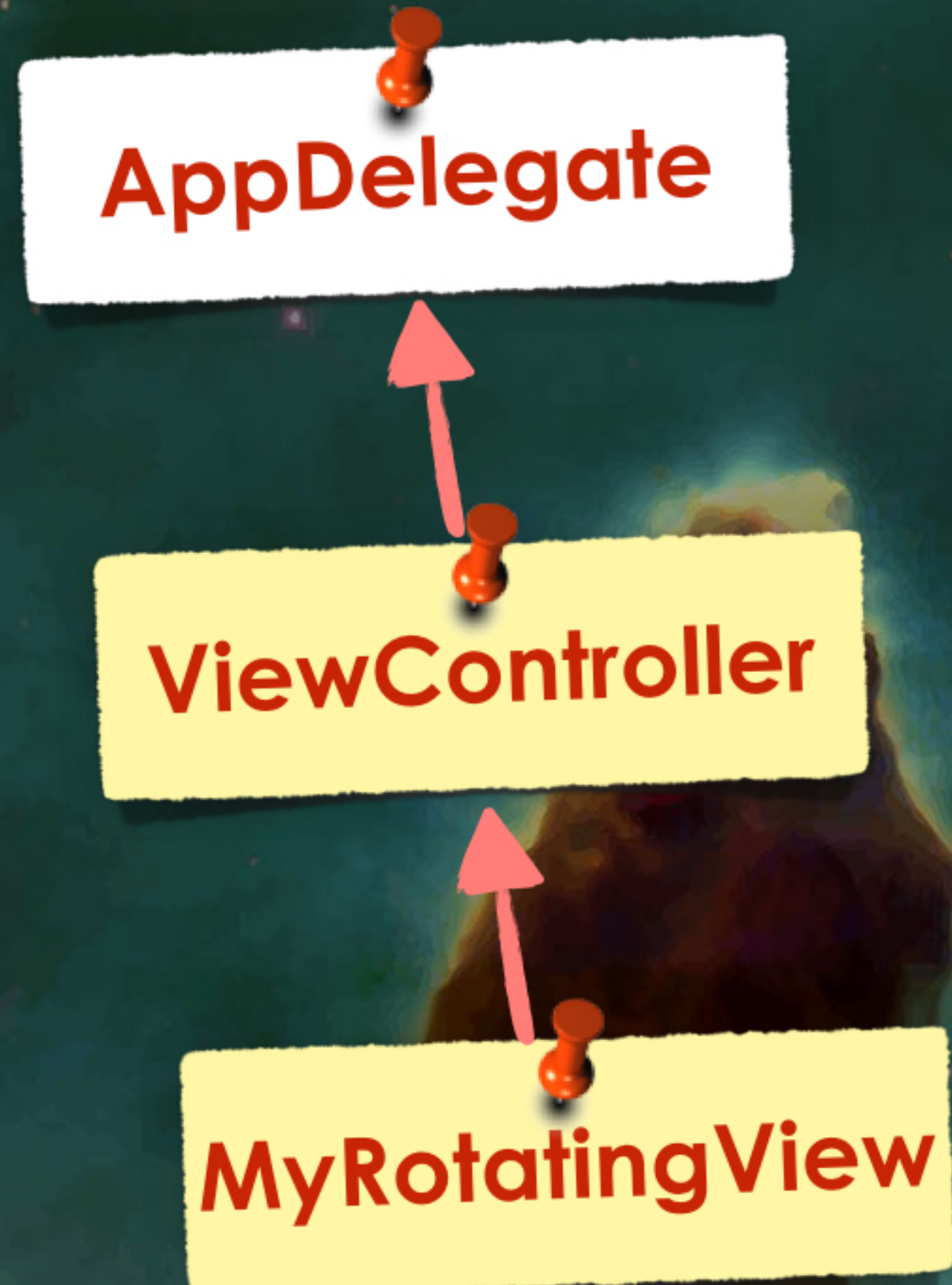
ViewController

- Handling button's action
- Handling orientation changes



MyRotatingView

- Display



Architecture

AppDelegate

- Do not touch (yet)

ViewController

- Handling button's action
- Handling orientation changes

MyRotatingView

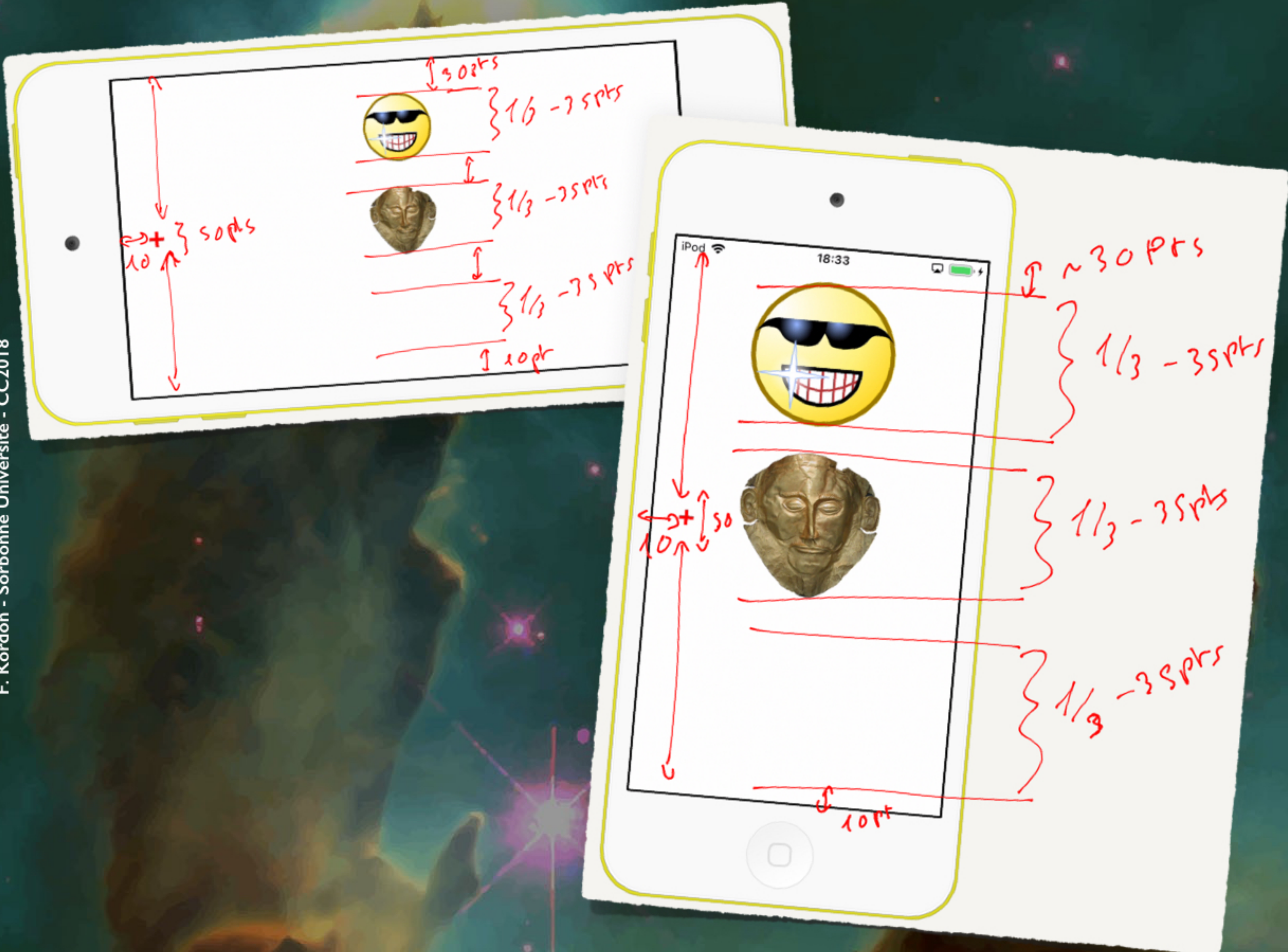
- Display



Strict MVC!

Despite no need for «model»

Drawings for this interface



ViewController (Objective-C)

```
//  
// ViewController.h  
// uRotate-objc  
//  
// Created by Fabrice Kordon on 28/09/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//
```

```
#import <UIKit/UIKit.h>
```

```
@interface ViewController : UIViewController
```

```
- (void) myAction:(UIButton*) sender;
```

```
@end
```

ViewController (Objective-C)

```
#import "ViewController.h"
#import "MyRotatingView.h"

@interface ViewController ()

@end

@implementation ViewController

MyRotatingView *v;

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    v = [[MyRotatingView alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    [v setBackgroundColor:[UIColor whiteColor]];
    [self setView:v];
    [v release];
}

- (void)viewWillTransitionToSize:(CGSize)size withTransitionCoordinator:
(id<UIViewControllerAnimatedTransitioning>)coordinator {
    [v drawInFormat:size]; // Implemented by MyRotatingView
}
```

ViewController (Objective-C)

```
- (void) myAction:(UIButton*) sender {
    if ([[v alien] isHidden]) {
        [[v alien] setHidden:NO];
        [[v smiley] setHidden:YES];
    } else {
        [[v alien] setHidden:YES];
        [[v smiley] setHidden:NO];
    }
    [v drawInFormat:[v bounds].size]; // Implemented by MyRoTaTingView
}

- (void) didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Nothing else to put here.
}

@end
```

ViewController (Swift)

```
//  
// ViewController.swift  
// uRotate-swift  
//  
// Created by Fabrice Kordon on 28/09/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        let v = MyRotatingView(frame: UIScreen.main.bounds)  
        self.view = v  
        v.backgroundColor = UIColor.white  
    }  
  
    override func viewWillTransition(to size: CGSize,  
        with coordinator: UIViewControllerTransitionCoordinator) {  
        super.viewWillTransition(to: size, with: coordinator)  
        let v = self.view as! MyRotatingView // avoids an error  
        v.drawInFormat(format:size) // Implemented by MyRoTaTingView  
    }  
}
```

ViewController (Swift)

```
// @objc to avoid a problem in addTarget
@objc func myAction (sender: UIButton!) {
    let v = self.view as! MyRotatingView // avoids an error
    if v.alien.isHidden {
        v.alien.isHidden = false
        v.smiley.isHidden = true
    } else {
        v.alien.isHidden = true
        v.smiley.isHidden = false
    }
    v.drawInFormat(format:v.bounds.size) // Implemented by MyRoTaTingView
}
```

MyRotatingView (Objective-C)

10

```
//  
// MyRotatingView.h  
// uRotate-objc  
//  
// Created by Fabrice Kordon on 28/09/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
#import <UIKit/UIKit.h>  
  
@interface MyRotatingView : UIView  
  
@property (readonly, nonatomic, retain) UIImageView *alien;  
@property (readonly, nonatomic, retain) UIImageView *smiley;  
  
- (void) drawInFormat: (CGSize) format;  
  
@end
```

MyRotatingView (Objective-C)

10

```
#import "MyRotatingView.h"  
#import "ViewController.h"
```

```
@implementation MyRotatingView
```

```
UIImageView *mask;  
CGFloat iconSize;  
UIButton *button;
```

MyRotatingView (Objective-C)

```
- (id) initWithFrame:(CGRect)frame {
    self = [super initWithFrame:frame];
    if (self) {
        _alien = [[UIImageView alloc]
            initWithImage:[UIImage imageNamed:@"alien"]];
        _smiley = [[UIImageView alloc]
            initWithImage:[UIImage imageNamed:@"smiley"]];
        mask = [[UIImageView alloc]
            initWithImage:[UIImage imageNamed:@"mask"]];
        button = [UIButton buttonWithType:UIButtonTypeSystem];
        [button setTitle:@"+" forState:UIControlStateNormal];
        [button setTitle:@" " forState:UIControlStateHighlighted];
        [_alien setHidden:YES];
        [[button titleLabel] setFont:[UIFont boldSystemFontOfSize:30.0]];
        [button setTintColor:[UIColor redColor]];
        [self addSubview:_alien];
        [self addSubview:mask];
        [self addSubview:_smiley];
        [self addSubview:button];
        [button addTarget:self.superview
            action:@selector(myAction:)
            forControlEvents:UIControlEventTouchUpInside];
        [_alien release];
        [mask release];
        [_smiley release];
        [self drawInFormat:[UIScreen mainScreen].bounds.size];
    }
    return self;
}
```


MyRotatingView (Objective-C)

10

```
- (void) drawInFormat: (CGSize) format {
    iconSize = format.height / 3 - 35.0;
    [_smiley setFrame:CGRectMake(format.width / 2.0 - (iconSize / 2.0),
                                30.0, iconSize, iconSize)];
    [mask setFrame:CGRectMake(format.width / 2.0 - (iconSize / 2.0),
                              format.height / 2.0 - (iconSize / 2.0),
                              iconSize, iconSize)];
    [_alien setFrame:CGRectMake(format.width / 2.0 - (iconSize / 2.0),
                                format.height - iconSize - 10.0,
                                iconSize, iconSize)];
    [button setFrame:CGRectMake(10.0, format.height / 2.0 - 25.0,
                               50.0, 50.0)];
}
@end
```

MyRotatingView (Swift)

11

```
//  
// MyRotatingView.swift  
// uRotate-swift  
//  
// Created by Fabrice Kordon on 28/09/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
import UIKit  
  
class MyRotatingView: UIView {  
  
    private var iconSize : CGFloat = 0.0  
  
    let alien = UIImageView(image : UIImage(named: "alien"))  
    let smiley = UIImageView(image : UIImage(named: "smiley"))  
    let theMask = UIImageView(image : UIImage(named: "mask"))  
    let button = UIButton(type: .system)
```

MyRotatingView (Swift)

```
override init (frame: CGRect) {
    button.setTitle("+", for: .normal)
    button.setTitle(" ", for: .highlighted)
    alien.isHidden = true
    button.tintColor = UIColor.red
    button.titleLabel!.font = UIFont.boldSystemFont(ofSize: 30.0)
    super.init(frame: frame); // You need self to be set-up now
    self.addSubview(alien)
    self.addSubview(theMask)
    self.addSubview(smiley)
    self.addSubview(button)
    button.addTarget(self.superview,
                    action: #selector(ViewController.myAction),
                    for: .touchUpInside)
    self.drawInFormat(format:frame.size)
}

required init?(coder aDecoder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
}
```

MyRotatingView (Swift)

```
func drawInFormat (format: CGSize) {  
    iconSize = format.height / 3 - 35.0 // mode paysage  
    smiley.frame = CGRect(x:format.width / 2.0 - (iconSize / 2.0),  
                           y:30.0, width:iconSize, height:iconSize)  
    theMask.frame = CGRect(x:format.width / 2.0 - (iconSize / 2.0),  
                            y:format.height / 2.0 - (iconSize / 2.0),  
                            width:iconSize, height:iconSize)  
    alien.frame = CGRect(x:format.width / 2.0 - (iconSize / 2.0),  
                          y:format.height - iconSize - 10.0,  
                          width:iconSize, height:iconSize)  
    button.frame = CGRect(x:10.0, y:format.height / 2.0 - 25.0,  
                           width:50.0, height:50.0)  
}
```

As a conclusion...

Honestly, is it complicated?

- Objective-C is more verbose
 - ▶ **An explicit management of memory (here)**
- No problem if you take care of it!

As a conclusion...

Honestly, is it complicated?

- Objective-C is more verbose
 - ▶ An explicit management of memory (here)
- No problem if you take care of it!



A few rules, you need to know

Rule 1

Before building you view,
its layout you will draw

Rule 2

From the size of the view,
coordinates or constraints
you will deduce

Could be of help...

Draw first with story board?

