



«Photographer»

Fabrice.Kordon@lip6.fr



Goal of the example

Play with an UIImagePickerController

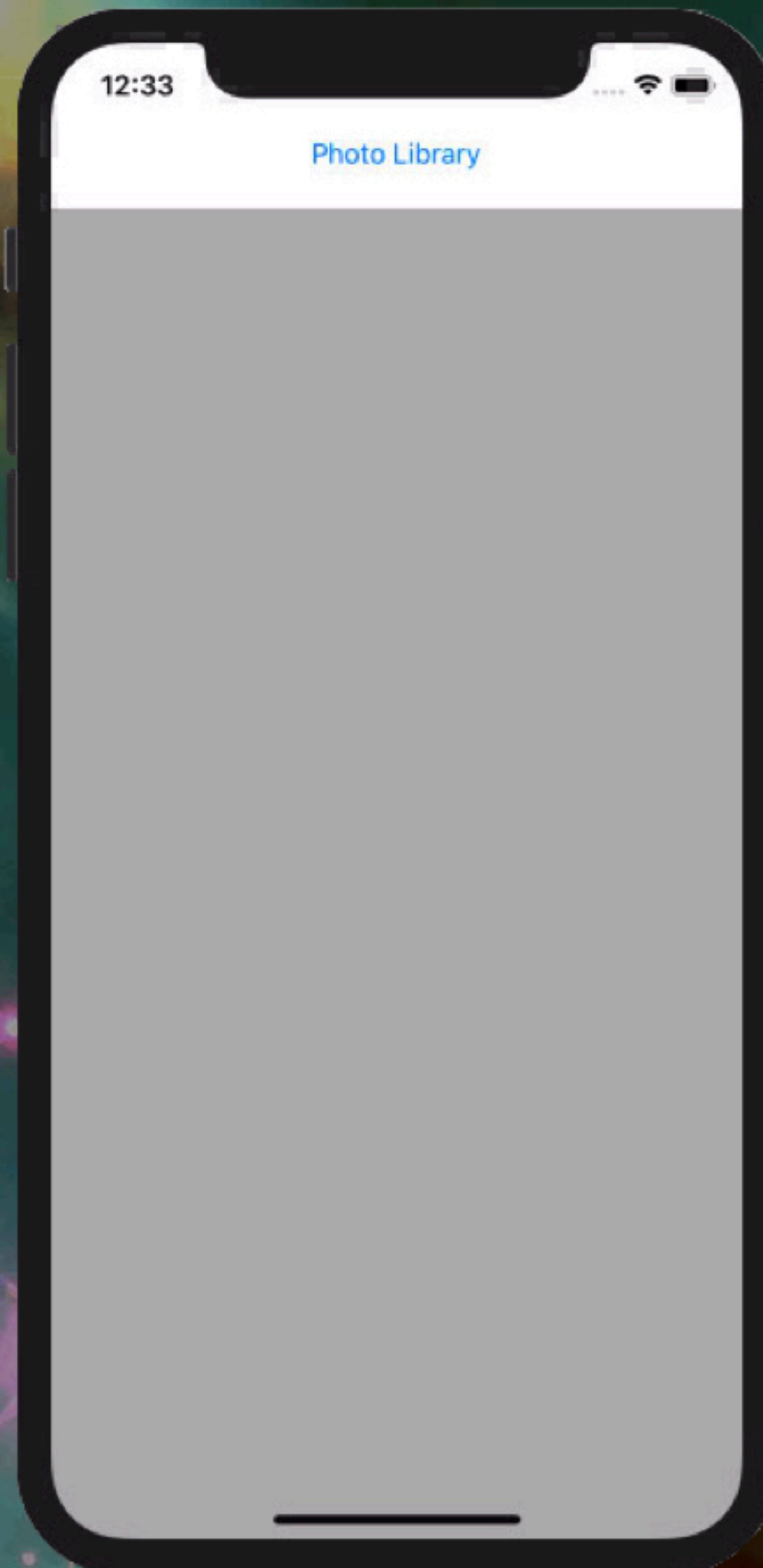
-  Take pictures...
-  Access to the photo library



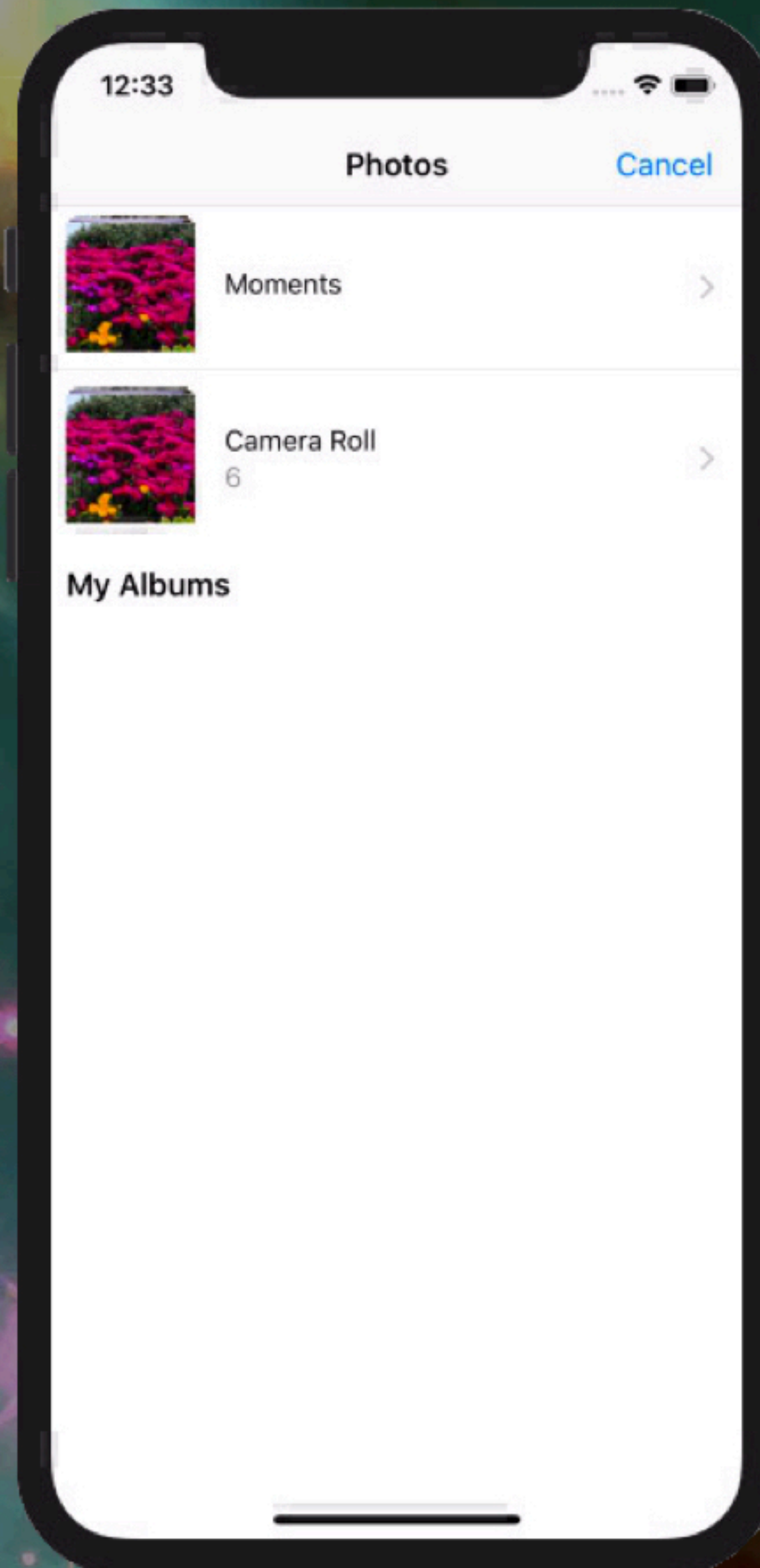
Demo



On the simulator?



On the simulator?



ViewController

```
//  
// ViewController.swift  
// Photographer  
//  
// Created by Fabrice Kordon on 20/10/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
import UIKit  
  
class ViewController: UIViewController {  
  
    let v = MyView(frame: UIScreen.main.bounds)  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        self.view = v  
    }  
  
    override func viewWillTransition(to size: CGSize,  
                                     with coordinator: UIViewControllerTransitionCoordinator) {  
        v.drawInFormat(size)  
    }  
}
```

MyView

```
import UIKit
```

```
class MyView: UIView, UINavigationControllerDelegate,  
             UIImagePickerControllerDelegate,  
             UIScrollViewDelegate {
```

```
    private let takePhoto = UIButton(type: .system)  
    private let photoLib = UIButton(type: .system)  
    private let scrollPict = UIScrollView()  
    private var aPicture = UIImageView()
```

MyView

```
override init(frame : CGRect) {
    super.init(frame: frame)
    self.backgroundColor = .white
    if UIImagePickerController.isSourceTypeAvailable(.camera) {
        takePhoto.setTitle("Take Photo", for: .normal)
        takePhoto.addTarget(self, action: #selector(doTakePhoto),
                            for: .touchDown)
        self.addSubview(takePhoto)
    }
    photoLib.setTitle("Photo Library", for: .normal)
    photoLib.addTarget(self, action: #selector(doSelectPhoto),
                      for: .touchDown)
    self.addSubview(photoLib)
    scrollPict.backgroundColor = .lightGray
    scrollPict.maximumZoomScale = 1.0
    scrollPict.minimumZoomScale = 0.05
    scrollPict.contentInset = UIEdgeInsets(top: 2, left: 2,
                                           bottom: 2, right: 2)
    scrollPict.delegate = self
    scrollPict.addSubview(aPicture)
    self.addSubview(scrollPict)
    self.drawInFormat(frame.size)
}

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


MyView

```
@objc func doTakePhoto() {  
    let imgPicker = UIImagePickerController()  
    imgPicker.delegate = self  
    imgPicker.sourceType = .camera  
    let vc = UIApplication.shared.windows[0].rootViewController  
    vc?.present(imgPicker, animated: true, completion: nil)  
}
```

```
@objc func doSelectPhoto() {  
    let imgPicker = UIImagePickerController()  
    imgPicker.delegate = self  
    imgPicker.sourceType = .photoLibrary  
    let vc = UIApplication.shared.windows[0].rootViewController  
    vc?.present(imgPicker, animated: true, completion: nil)  
}
```

MyView

```
func drawInFormat(_ s: CGSize) {
    var top = 20;
    if UIDevice.current.userInterfaceIdiom == .phone &&
        s.height >= 812 {
        top = 30
    } else if UIDevice.current.userInterfaceIdiom == .phone &&
        s.width > s.height {
        top = 0
    }
    // Rare since now, no running device should be without camera
    if UIImagePickerController.isSourceTypeAvailable(.camera) {
        takePhoto.frame = CGRect(x: Int(s.width / 4 - 50),
            y: top + 20, width: 100, height: 20)
        photoLib.frame = CGRect(x: Int(s.width / 4 * 3 - 50),
            y: top + 20, width: 100, height: 20)
    } else {
        photoLib.frame = CGRect(x: Int(s.width / 2 - 50),
            y: top + 20, width: 100, height: 20)
    }
    scrollPict.frame = CGRect(x: 0, y: top + 60, width: Int(s.width),
        height: Int(s.height - 80))
}
```

MyView

```
// UIImagePickerControllerDelegate protocol

func imagePickerControllerDidCancel(_ picker: UIImagePickerController) {
    picker.dismiss(animated: true, completion: nil)
}

func imagePickerController(_ picker: UIImagePickerController,
    didFinishPickingMediaWithInfo
        info: [UIImagePickerController.InfoKey : Any]) {
    picker.dismiss(animated: true, completion: nil)
    aPicture.removeFromSuperview()
    let img = info[UIImagePickerController.InfoKey.originalImage] as! UIImage
    aPicture = UIImageView(image:img)
    scrollPict.addSubview(aPicture)
    scrollPict.setNeedsDisplay()
}


// UIScrollViewDelegate protocol

func viewForZooming(in scrollView: UIScrollView) -> UIView? {
    return aPicture
}

func scrollViewDidEndZooming(_ scrollView: UIScrollView,
    with view: UIView?,
    atScale scale: CGFloat) {
    scrollView.zoomScale = scale
}
}
```

As a conclusion...

Nothing difficult is'n't it?

-  Do not forget the info.plist declaration
-  And only declare what you need...

Some more nice stuff to enhance your Apps

