AOP在iOS App中的运用

AOP的优点

1. 不侵入原有代码
2. 不影响继承体系

OC和Swift的支持

AOP依赖语言原生支持，或者强大的runtime。

OC虽然在语言层面不像Java那样支持AOP，但是因为有强大的runtime，所以可以实现AOP。

Aspects是一个使用非常广泛的开源库。支持before、instead、after三种hook方式，并且同一个拦截点支持设置多个block。

Swift的runtime非常简单，目前还不支持AOP。因为UIKit目前还是OC写的，所以Swift工程也能使用Aspects。

阿里云App的特点

1. 完全基于Swift 3.0开发。
2. 实现组件化，每个组件都有自己的git。如果一个需求牵涉太多组件，做起来会非常艰难。

AOP使用的几个场景

1. 实现视觉定制化，比如埋点、定制VC返回按钮的样式。
2. 实现特定业务，比如宝箱。
3. 开发版往控制台打印日志。

视觉定制化

我们VC的返回按钮没有采用标准的样式，所以要达成统一的样式，传统的做法是在基类里面做好，然后大家都从这个基类派生。

这样的做法影响比较大，对第三方引入的组件无效，比如扫码SDK。

视觉定制化

使用Aspects hook住VC几个显示函数，进行是视觉的设置。

```
public func aly_viewDidLoad() -> Void {
    self.navigationController?.navigationBar.barStyle = .blackTranslucent

    self.view.backgroundColor = UIColor.white
    if self.aly_showBackBarButton == true && self.navigationController != nil {
        self.navigationItem.leftBarButtonItem = ALYUIUtils.createBarButtonItem(self, iconName: "bar_back_icon",
                                pressedIcon: "bar_back_icon",
                                title: "",
                                action: #selector(handleNavigationBackAction(_:)),
                                withFrame: CGRect(x: 0, y: 0, width: 22, height: 22),
                                offsetX: 6,
                                withUserdata: nil)
    }

    self.alyLifeCycleHookBridge(String(describing: type(of: self)), selectorName: "viewDidLoad")
}
```

宝箱活动

运营活动，进入某些页面，要显示宝箱，用户点击后可以抽奖。

页面列表配置在服务器端，可以动态更新。

使用AOP hook vc相关显示函数，获取到VC的名字跟列表比对，如果匹配上了，就将宝箱显示出来。

打印日志

develop版要往控制台打印日志，正式版则只能往文件里面打印日志，所以在ALYDebugKit里面hook打印日志的函数，往控制台打印日志。

```
let actionBlock:@convention(block) (AspectInfo)-> Void = { (aspectInfo) -> Void in

    if let message = aspectInfo.arguments()[0] as? String {
        NSLog(message)
    }
}

let wrappedBlock: AnyObject = unsafeBitCast(actionBlock, to: AnyObject.self)

do {
    try ALYLogger.aspect_hook(#selector(ALYLogger.log),
                              with: [],
                              usingBlock: wrappedBlock)
} catch let error as NSError {
    NSLog(error.description)
    abort()
}
```

Aspects 跟 JSPatch冲突问题

JSPatch跟Aspects都是iOS App里面使用非常广泛的开源组件，但是不幸的是，因为二者原理想死，如果同时替换同一个函数，有可能会导致冲突。

全面谈谈Aspects和JSPatch兼容问题（更新建议）
http://www.jianshu.com/p/dc1deaa1b28e

我们的办法

不修改Aspects和JSPatch的源代码。

```
class func hook_viewDidDisappear() -> Void {
    let actionBlock:@convention(block) (AspectInfo)-> Void = { aspectInfo in
        if let instance = aspectInfo.instance() as? UIViewController {
            let className = String(describing: type(of: instance))
            if UIViewController.aly_shouldHookViewController(className) {
                UTAnalytics.getInstance().getDefaultTracker().pageDisAppear(instance)
                instance.alyLifeCycleHookBridge(String(describing: type(of: instance)), selectorName: "viewDidDisappear")
            }
        }
    }

    let wrappedBlock: AnyObject = unsafeBitCast(actionBlock, to: AnyObject.self)

    do {
        try UIViewController.aspect_hook(#selector(UIViewController.viewDidDisappear(_:)), with: ALYVCHookPositionAfter, usingBlock:
        wrappedBlock)
    } catch let error as NSError {
        ALYLogError(error.description, module: "", category: .Exceptoin)
    }
}
```

```
public dynamic func alyLifeCycleHookBridge(_ className: String, selectorName: String) {

}
```