

设计一套iOS暗黑适配组件

- ◆ iOS 关联对象
- ◆ iOS交换方法（黑魔法）

2021年7月 DevDragonLi

暗黑模式下原生和H5界面适配样例效果



产品层需求汇总

- ◆ 适配APP的原生界面。
- ◆ 前端界面的适配。
- ◆ 动态模式，也可自定义开关状态。

端上技术层面分析

- ◆ 仅 iOS 13 可实现跟随系统的暗黑模式 【iOS13以下需要组件兼容】
- ◆ 对于APP进入后台等行为，iOS系统会给出通知，但暗黑模式切换并未给出通知，仅有一个UIColor的API,需要返回二个颜色值
- ◆ 对于CGColor类型，iOS并无动态API适配，需要切换状态时，再次设置颜色值。

iOS Dark Mode API (1)

```
118 + (UIColor *)colorWithDynamicProvider:(UIColor * (^)(UITraitCollection
    *traitCollection))dynamicProvider API_AVAILABLE(ios(13.0), tvos(13.0))
    API_UNAVAILABLE(watchos);
119 - (UIColor *)initWithDynamicProvider:(UIColor * (^)(UITraitCollection
    *traitCollection))dynamicProvider API_AVAILABLE(ios(13.0), tvos(13.0))
    API_UNAVAILABLE(watchos);
```

```
1 *   userInterfaceIdiom, userInterfaceStyle, displayGamut, accessibilityContrast,
    userInterfaceLevel
2 * and more could be added in the future.
3 */
4 - (BOOL)hasDifferentColorAppearanceComparedToTraitCollection:(nullable
    UITraitCollection *)traitCollection API_AVAILABLE(ios(13.0), tvos(13.0))
    API_UNAVAILABLE(watchos);
```


iOS Dark Mode API (2)

```
- (void)traitCollectionDidChange:(UITraitCollection *)previousTraitCollection {
    [super traitCollectionDidChange:previousTraitCollection];

    if ([self.traitCollection hasDifferentColorAppearanceComparedToTraitCollection:
        // Config Color Adapter
    ]) {
    }
}
```

VC Config Single Style

```
if (@available(iOS 13.0, *)) {

    [self setOverrideUserInterfaceStyle:UIUserInterfaceStyleDark];
}
```

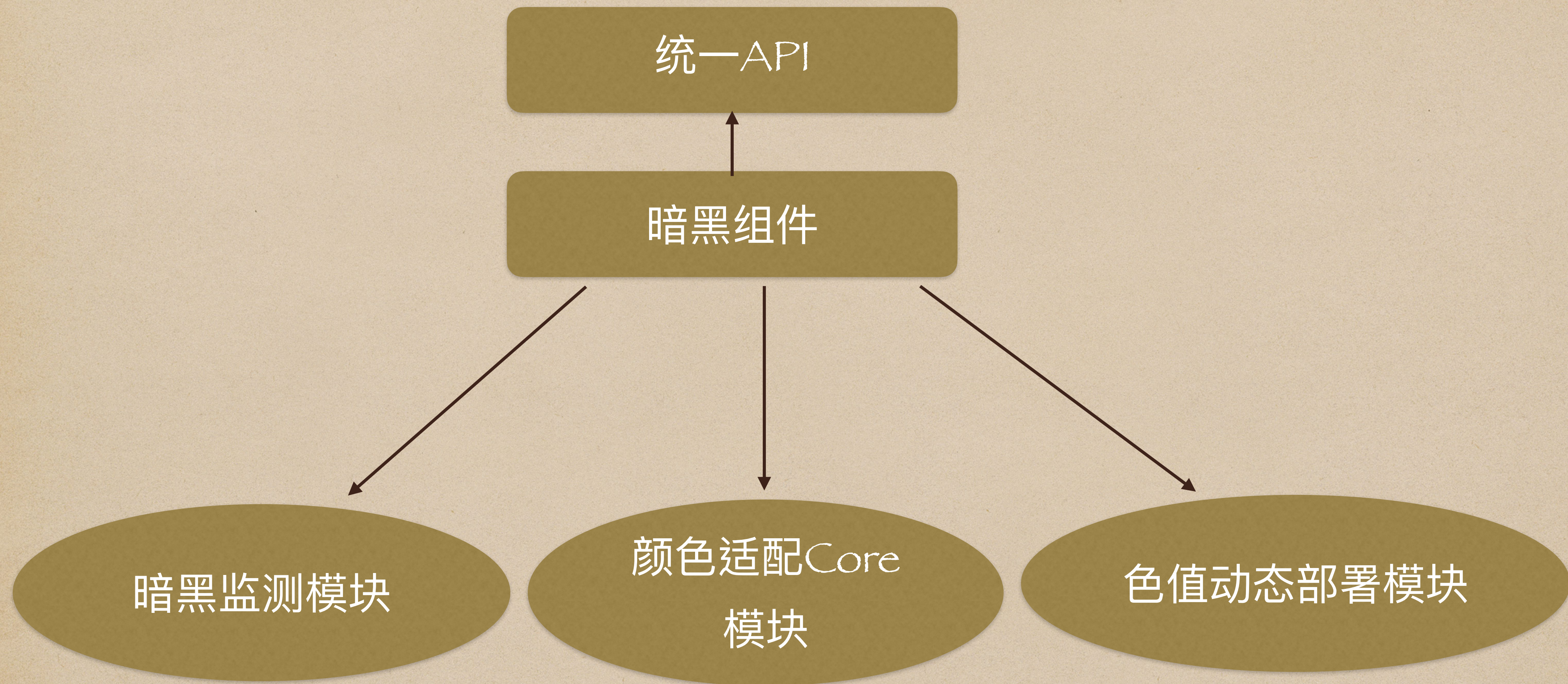
Life Style

- UIView
 - traitCollectionDidChange(_:)
 - layoutSubviews()
 - draw(_:)
 - updateConstraints()
 - tintColorDidChange()
- UIViewController
 - traitCollectionDidChange(_:)
 - updateViewConstraints()
 - viewWillLayoutSubviews()
 - viewDidLayoutSubviews()
- UIPresentationController
 - traitCollectionDidChange(_:)
 - containerViewWillLayoutSubviews()
 - containerViewDidLayoutSubviews()

组件技术指标

- ◆ 既可以适配iOS13以后的设备，也可以让iOS13以下设备体验新的适配界面（APP）。
- ◆ 解决`暗黑模式`开关切换的状态同步问题。
- ◆ 对于UIColor和CGColor 可以实现一行代码配置，RD无需关心各类回调函数内繁琐适配及各类逻辑。
- ◆ 较低适配成本。

组件模块结构



版本适配

- ◆ 设备为iOS13+，则用户可选跟随系统设置，或者用户自主开关
- ◆ 设备为iOS13以下，则用户可选自主开关控制显示模式，此模式下需要组件内部维护开关

端上暗黑状态开关图示



暗黑状态监测同步(1)

```
7 if (@available(iOS 13.0, *)) {
8     UIColor *dynamicColor = [UIColor colorWithDynamicProvider:^(UIColor *
9         _Nonnull(UITraitCollection * _Nonnull traitCollection) {
10         if (traitCollection.userInterfaceStyle == UIUserInterfaceStyleDark) {
11             self.darkModeStyle = YES;
12             [self postDarkModeChangeNotificationName];
13             [self updateMonitorStatus];
14             NSLog(@"\n😁 DarkMode Now ! \n😁 【LFLDarkModeKitTips Debug Msg】 ");
15             return [UIColor blackColor];
16         } else {
17             self.darkModeStyle = NO;
18             [self postDarkModeChangeNotificationName];
19             [self updateMonitorStatus];
20             NSLog(@"\n😁 LightMode Now ! \n😁 【LFLDarkModeKitTips Debug Msg】 ");
21             return [UIColor grayColor];
22         }
23     }];
24     UIView *unVisibleView = [[UIView alloc] initWithFrame:CGRectZero];
25     unVisibleView.backgroundColor = dynamicColor;
26     unVisibleView.hidden = YES;
```


暗黑状态监测同步(2)

- ◆ 如果是中小型项目，可通过设置系统window颜色为动态，即可实现全局监控托管。
- ◆ 基础组件模式：通过一个不可见View 设置动态颜色，来解决监测问题（内部可抛出通知来发布状态）

UIColor及CGColor适配

- ◆ 通过hook `environmentTraitCollectionDidChange` 函数，内部处理 状态切换的颜色动态改变。
- ◆ 具体可参考：UIView+LFLDarkMode.h 【Line 26 ~ line 47】

色值适配

- ◆ 色值资源管理：双字典Plsit文件包装为Bundle资源文件
- ◆ 通过资源文件描述Light 和Dark模式下颜色对应色值的字典维护
(Example：Key：PGC01 Light Mode Value：000000 Dark Mode Value：FFFFFF)
- ◆ 组件内部通过开关读取对应的色值
- ◆ 系统状态开关切换，实现上述hook函数的内部切换逻辑。

How To Use DarkModeKit

```
8 pod 'LFLDarkModeKit', '~>3.2.0'
```

```
21 NSURL *darkModeBundleURL = [[NSBundle mainBundle]
    URLForResource:@"darkModeAdapterColor" withExtension:@"bundle"];
22 [[LFLDarkModeManger sharedInstance]
    configDarkModeColorBundleURL:darkModeBundleURL];
23 [LFLDarkModeManger.sharedInstance configUserDarkMode:NO];
```

```
[[NSNotificationCenter defaultCenter] addObserver:self
    selector:@selector(_darkModeNoti:) name:LFLDarkModeChangeNotification
    object:nil];
```

```
UIColor *dyColor = [UIColor colorAdpterWithHex:LFLDarkModeTool.PColor0String];
self.exampleButton.backgroundColor = dyColor;
self.customView.layerBorderColorHex = [LFLDarkModeTool PColor0String];
```


QA

- ◆ 系统提供了颜色的适配方式，为何还需要再设计一套？
- ◆ 图片的适配方案 【向下兼容策略】
- ◆ 基本实现预期效果，但是依旧存在CGColor存在侵入API问题[待改进]

Last Part

- ◆ Apple : <https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/dark-mode/>
- ◆ GitHub : <https://github.com/DevDragonLi/LFLDarkModeKit>

谢谢观看