统一弹窗视图控制器

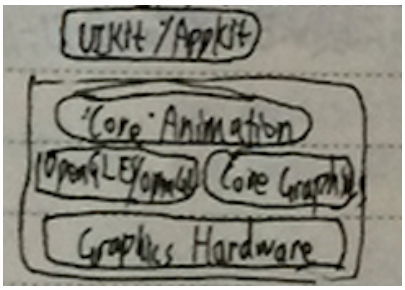
1. 简述：

项目中有许多弹窗，如分享、二维码等，其中必然包含很多逻辑，可在视图控制器中处理(不要在 view 层处理业务逻辑)；把 tipsView 改为 tipsViewController 实现一个 BaseViewController，弹窗动画和背景模糊在 Base 中配置。

2. 实现：

(1) 先介绍下 coreAnimation 动画相关的一些基础知识

1) ios 中，图形可以分为以下几个层次：



越往上，封装度越高，自由度越低，但使用越简单。

2) ios 中，展示 coreAnimation 动画类似于“拍电影”。三大要素如下：

演员->CALayer，规定电影主角是谁

剧本->CAAnimation，规定电影怎么演，怎么走，怎么变换

开拍->AddAnimation，开始执行

3) CALayer 是个与 UIView 类似的概念，同样有 layer，sublayer...，同样有 backgroundColor、frame 等相似属性，可将 UIView 视为一种特殊的 CALayer，只不过 UIView 可以响应事件而已，一般 layer 有两种用途：一是对 view 相关属性的设置，包括圆角阴影边框等，二是实现对 view 的动画操作。因此，对一个 view 进行 animation 动画，本质是对该 view 的 layer 进行动画操作。

4) CAAnimation 可分为四种

CABasicAnimation：通过设置起始点、终点、时间，动画会沿着设定点进行移动，可视为特殊的 CAKeyframeAnimation

CAKeyframeAnimation：关键点点 frame，可通过设定 CALayer 的始中末点 frame、时间，动画会沿着设定轨迹进行移动

CAAnimationGroup：也就是组合动画，把对这个 Layer 的所有动画都组合起来，一个 Layer 设定很多动画可以同时执行也可以顺序执行

CATransition：苹果帮开发者封装好的一些动画

(2) BaseTipsViewController 的实现

```
@interface MYTipsCoverView : UIView

@property (nonatomic, assign) BOOL isBlur;
@property (nonatomic, assign) CGFloat blurRadius;
@property (nonatomic, strong) UIColor *blurColor;

- (id)initWithBlur:(BOOL)isBlur;

@end
```

```
@implementation MYTipsCoverView

- (instancetype)init {
    if (self = [super init]) {
        self.frame = [[UIScreen mainScreen] bounds];
        _bg = [[UIImageView alloc] initWithFrame:self.frame];
        [self.layer addSublayer:_bg.layer];
    }
    return self;
}

- (id)initWithBlur:(BOOL)isBlur {
    self.isBlur = isBlur;
    self = [super init];
    if (self) {
        self.blurRadius = 6.0f;
        self.blurColor = [UIColor colorWithWhite:0 alpha:0.5];
        [self performScreenshotAndBlur];
    }
    return self;
}
```

```
- (id)initWithFrame:(CGRect)frame {
    self = [super initWithFrame:[[UIScreen mainScreen] bounds]];

    if (self) {
        _bg = [[UIImageView alloc] initWithFrame:self.frame];
        [self addSubview:_bg];
        self.blurRadius = 6.0f;
        self.blurColor = [UIColor colorWithWhite:0 alpha:0.5];
        [self performScreenshotAndBlur];
    }

    return self;
}
```

```
#pragma mark private Method

-(UIImage *)convertViewToImage {
    UIWindow *keyWindow = [[UIApplication sharedApplication] keyWindow];
    CGRect rect = [keyWindow bounds];
    UIGraphicsBeginImageContextWithOptions(rect.size,YES, 0.0f);
    CGContextRef context = UIGraphicsGetCurrentContext();
    [keyWindow.layer renderInContext:context];
    UIImage *capturedScreen = UIGraphicsGetImageFromCurrentImageContext();
    UIGraphicsEndImageContext();
    return capturedScreen;
}
```

```objc
- (void)performScreenshotAndBlur {
    UIImage *bgImage = [self convertViewToImage];
    if (self.isBlur) {
        bgImage = [bgImage applyBlurWithRadius:self.blurRadius tintColor:self.blurColor saturationDeltaFactor:1.0 maskImage:nil];
    } else {
        bgImage = [bgImage applyBlurWithRadius:0 tintColor:self.blurColor saturationDeltaFactor:1.0 maskImage:nil];
    }
    [_bg setImage:bgImage];
    _bg.alpha = 1.0f;
}

- (void)setIsBlur:(BOOL)isBlur {
    _isBlur = isBlur;
    [self performScreenshotAndBlur];
}

- (void)setBlurColor:(UIColor *)blurColor {
    _blurColor = blurColor;
    [self performScreenshotAndBlur];
}

- (void)setBlurRadius:(CGFloat)blurRadius {
    _blurRadius = blurRadius;
    [self performScreenshotAndBlur];
}
```

```objc
@interface MYBaseTipsViewController : UIViewController

/**
 * 是否需要模糊效果
 */
@property (nonatomic, assign) BOOL isBlur;

/**
 * 模糊程度
 */
@property (nonatomic, assign) CGFloat blurRadius;

/**
 * 背景色 （PS：最好带透明度）
 */
@property (nonatomic, strong) UIColor *blurColor;

/**
 * 背景是否接受点击事件
 */
- (void)setBackgroundEnable:(BOOL)enable;

/**
 * 插入到parentViewController
 *
 * @param superViewController parentViewController
 */
- (void)insertIntoParentViewController:(UIViewController *)parentViewController;
```

```objc
/**
 * 移除TipsViewController，默认动画
 */
- (void)onCloseWithAnimationDuration:(CGFloat)duration;

/**
 * 替换背景点击事件移除TipsViewController，自定义动画
 */
- (void)setTarget:(id)target tapSEL:(SEL) tapAction ;

/**
 * 中间展示视图是否需要动画
 *
 * @param animationView 中间展示视图
 */
- (void)showViewWithAnimation:(UIView *)animationView;
```

```objc
#import "MYBaseTipsViewController.h"
#import "MYTipsCoverView.h"

@interface MYBaseTipsViewController ()

@property (nonatomic, strong) UIView *middleView;

@property (nonatomic, strong) MYTipsCoverView *backgroundView;
@property (nonatomic, strong) UITapGestureRecognizer *tapGesture;

@end

@implementation MYBaseTipsViewController

- (MYTipsCoverView *)backgroundView {
    if (!_backgroundView) {
        _backgroundView = [[MYTipsCoverView alloc] initWithBlur:self.isBlur];
    }
    return _backgroundView;
}

- (void)viewDidLoad {
    [super viewDidLoad];
    self.backgroundView.userInteractionEnabled = YES;
    _tapGesture = [[UITapGestureRecognizer alloc] initWithTarget:self action:@selector(onBackgroundClicked)];
    _tapGesture.numberOfTapsRequired = 1;
    [self.backgroundView addGestureRecognizer:_tapGesture];
}
```

```objc
- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];
    [self.view setBackgroundColor:[UIColor clearColor]];
    [self.view insertSubview:self.backgroundView atIndex:0];
    self.backgroundView.alpha = 0;
    __weak typeof(self) weakSelf = self;
    [UIView animateWithDuration:0.2 delay:0 options:UIViewAnimationOptionCurveEaseOut animations:^{
        weakSelf.backgroundView.alpha = 1;
    } completion:^(BOOL finished) {

    }];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
```

```objc
- (void)showViewWithAnimation:(UIView *)animationView {
    self.middleView = animationView;
    animationView.alpha = 0.0f;

    __weak typeof(self) selfWeak = self;
    [UIView animateWithDuration:0.05 delay:0.0 options:UIViewAnimationOptionCurveEaseOut animations:^{
        selfWeak.backgroundView.alpha = 1.0;
    } completion:^(BOOL finished) {
        CABasicAnimation *scaleAnimation = [CABasicAnimation animationWithKeyPath:@"transform.scale"];
        scaleAnimation.fromValue = [NSNumber numberWithFloat:1.0];
        scaleAnimation.toValue = [NSNumber numberWithFloat:0.0];
        scaleAnimation.fillMode = kCAFillModeForwards;
        scaleAnimation.removedOnCompletion = NO;
        scaleAnimation.duration = 0.05;
        scaleAnimation.delegate = self;
        [animationView.layer addAnimation:scaleAnimation forKey:@"scales"];
    }];
}
```

```
- (void)animationDidStop:(CAAnimation *)anim finished:(BOOL)flag {
    self.middleView.alpha = 1.0f;
    CAKeyframeAnimation *animation = [CAKeyframeAnimation animationWithKeyPath:@"transform.scale"];
    animation.duration = 0.2;
    animation.values = @[@0, @0.41, @0.82, @1.0];
    animation.keyTimes = @[@0, @(3.0/6.0), @(5.0/6.0), @1];
    animation.additive = YES;
    animation.fillMode = kCAFillModeForwards;
    animation.removedOnCompletion = NO;
    [self.middleView.layer addAnimation:animation forKey:@"scale"];
}

- (void)insertIntoParentViewController:(UIViewController *)parentViewController {
    [parentViewController addChildViewController:self];
    [self.view setTranslatesAutoresizingMaskIntoConstraints:NO];
    [parentViewController.view addSubview:self.view];

    NSDictionary *layoutViews = @{@"view":self.view};
    NSArray *constraints_H = [NSLayoutConstraint constraintsWithVisualFormat:@"H:|[view]|" options:0 metrics:nil views:layoutViews];
    NSArray *constraints_V = [NSLayoutConstraint constraintsWithVisualFormat:@"V:|[view]|" options:0 metrics:nil views:layoutViews];
    [parentViewController.view addConstraints:constraints_H];
    [parentViewController.view addConstraints:constraints_V];

    [self willMoveToParentViewController:parentViewController];
    [self beginAppearanceTransition:YES animated:NO];
    [self endAppearanceTransition];
}
```

```
- (void)onCloseWithAnimationDuration:(CGFloat)duration {
    if(duration < 0)
        duration = 0.33;
    __weak typeof(self) selfWeak = self;
    [UIView animateWithDuration:duration
                          delay:0.0
                        options: UIViewAnimationOptionCurveEaseInOut
                     animations:^{
                         selfWeak.view.alpha = 0;
                     }
                     completion:^(BOOL finished){
                         [selfWeak willMoveToParentViewController:nil];
                         [selfWeak.view removeFromSuperview];
                         [selfWeak removeFromParentViewController];
                     }];
}

- (void)onBackgroundClicked{
    [self onCloseWithAnimationDuration:0.33];
}

- (void)setBlurRadius:(CGFloat)blurRadius {
    _blurRadius = blurRadius;
    self.backgroundView.isBlur = YES;
    self.backgroundView.blurRadius= blurRadius;
}
```

```
- (void)setBlurColor:(UIColor *)blurColor {
    _blurColor = blurColor;
    self.backgroundView.isBlur = YES;
    self.backgroundView.blurColor = blurColor;
}

- (void)setBackgroundEnable:(BOOL)enable {
    self.backgroundView.userInteractionEnabled = enable;
}

- (void)setTarget:(id)target tapSEL:(SEL)tapAction{
    [self.tapGesture removeTarget:self action:@selector(onBackgroundClicked)];
    [self.tapGesture addTarget:target action:tapAction];
}
```