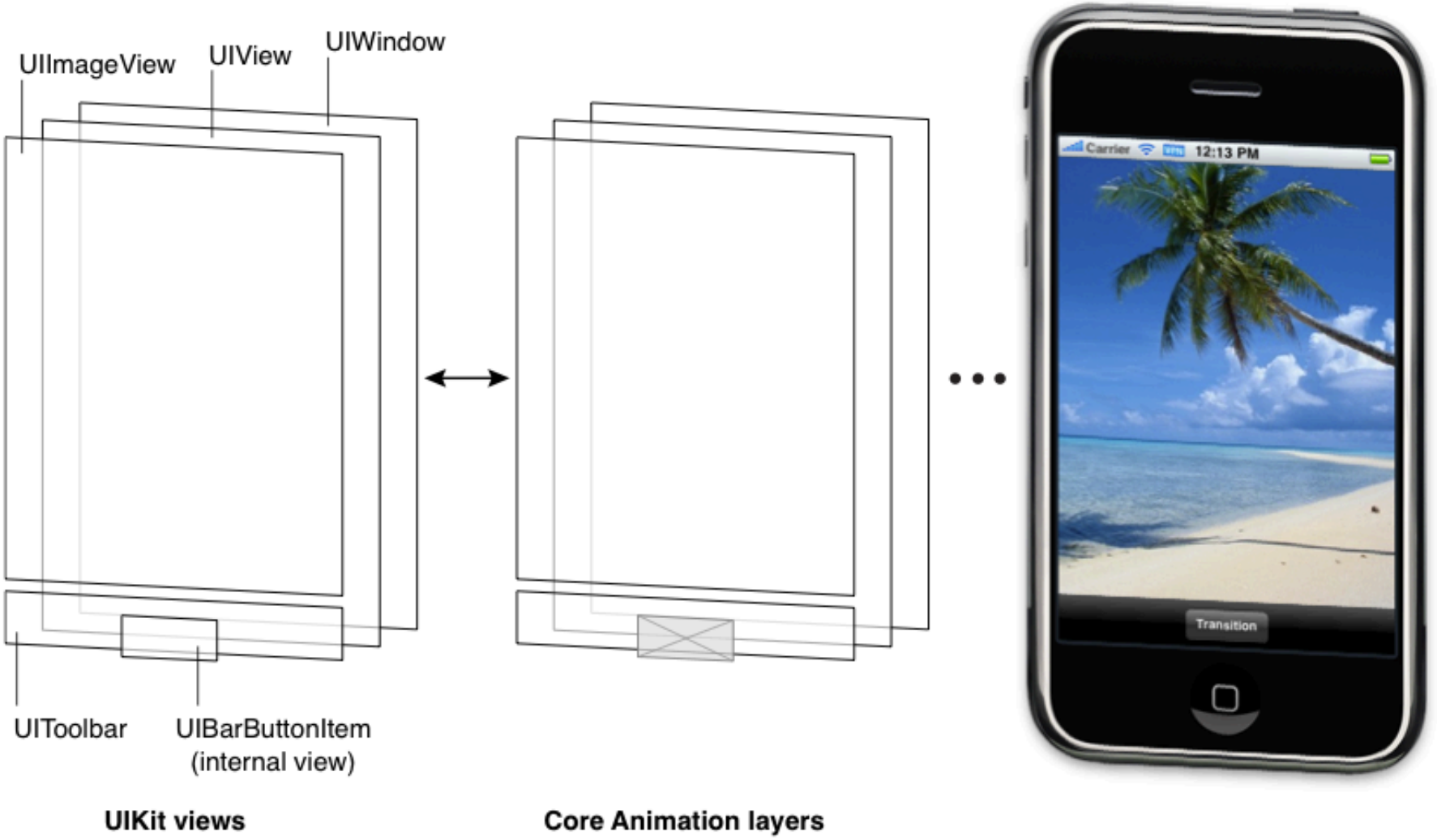
UIView


UIImageView
UIView
UIWindow
UIToolbar
UIBarButtonItem
(internal view)
Carrier
12:13 PM
Transition
UIKit views
Core Animation layers

Liên quan đến UIView

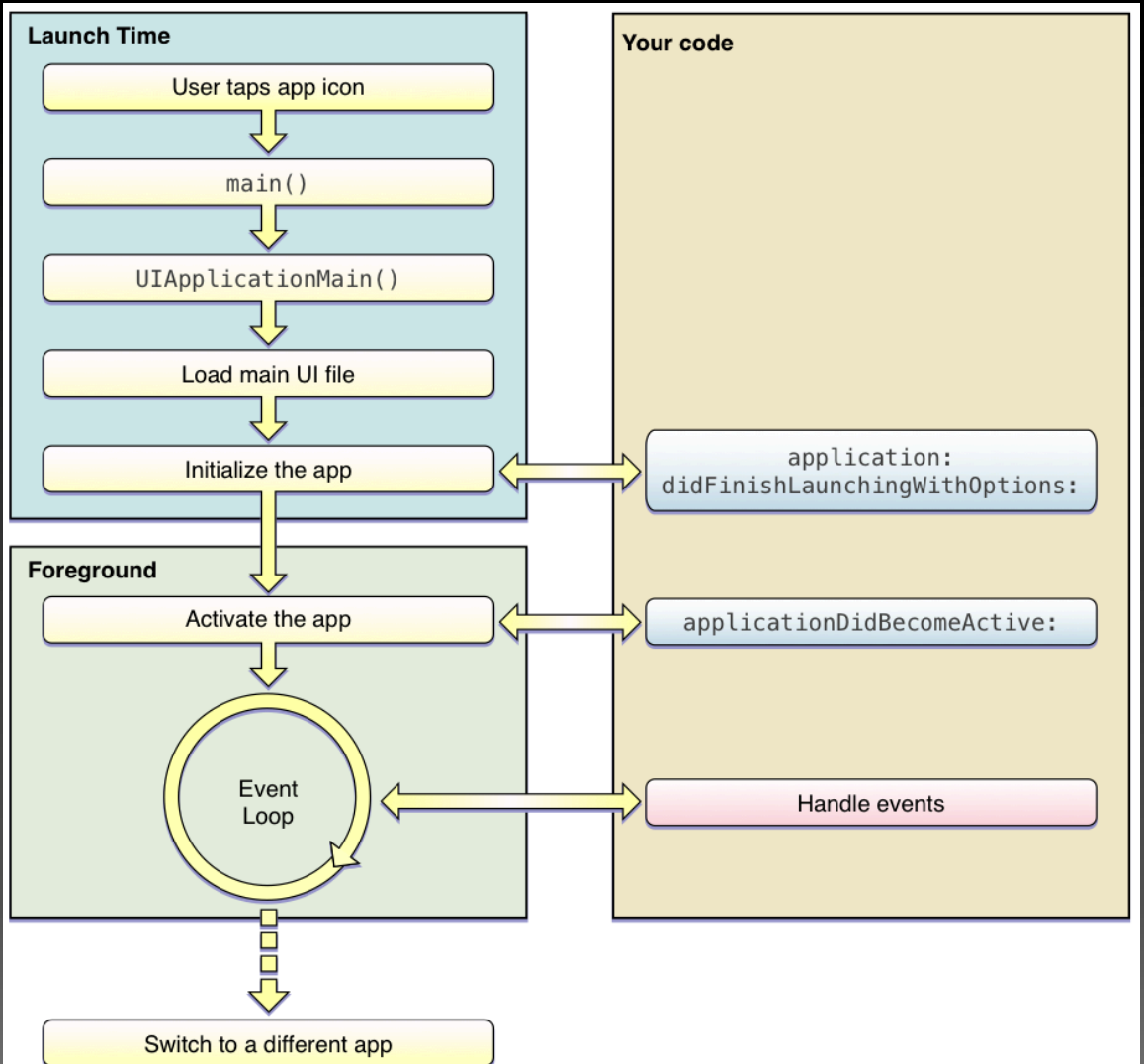
- Hierarchy (bố, con, anh, chị, trên, dưới)
- Geometry (ở chỗ nào, to, bé ra làm sao)
- Rendering (hiển thị đậm nhạt, ẩn hiện…)
- Animation (hoạt hình cổ điển)
- Animation with Block (hoạt hình sử dụng block)

Quan hệ giữa UIView và UIViewController

- Thường là 1:1, tuy nhiên một UIViewController có thể load > 1 UIView từ NIB file
- Viết mã logic trong UIViewController để điều khiển UIView
- Subclass UIView
- Tạo category UIView hoặc biến thể của nó

Cách ứng dụng khởi tạo giao diện có duy nhất một ViewController

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]];

    // Override point for customization after application launch.
    self.viewController = [[ViewController alloc]
initWithNibName:@"ViewController" bundle:nil];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}
```



Sự kiện trong vòng đời của UIView trong UIViewController

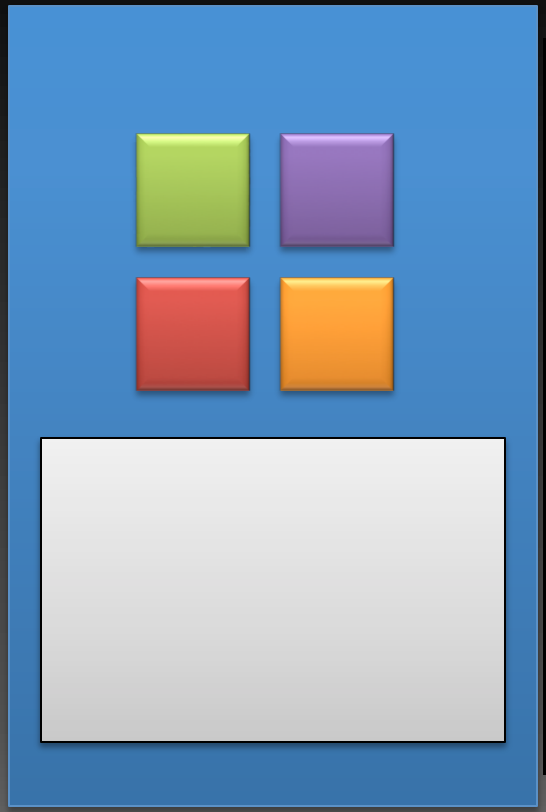
- ViewDidLoad > ViewWillAppear > ViewDidAppear
- ViewWillDisappear > ViewDidDisappear > View > ViewWillUnload > ViewDidUnload

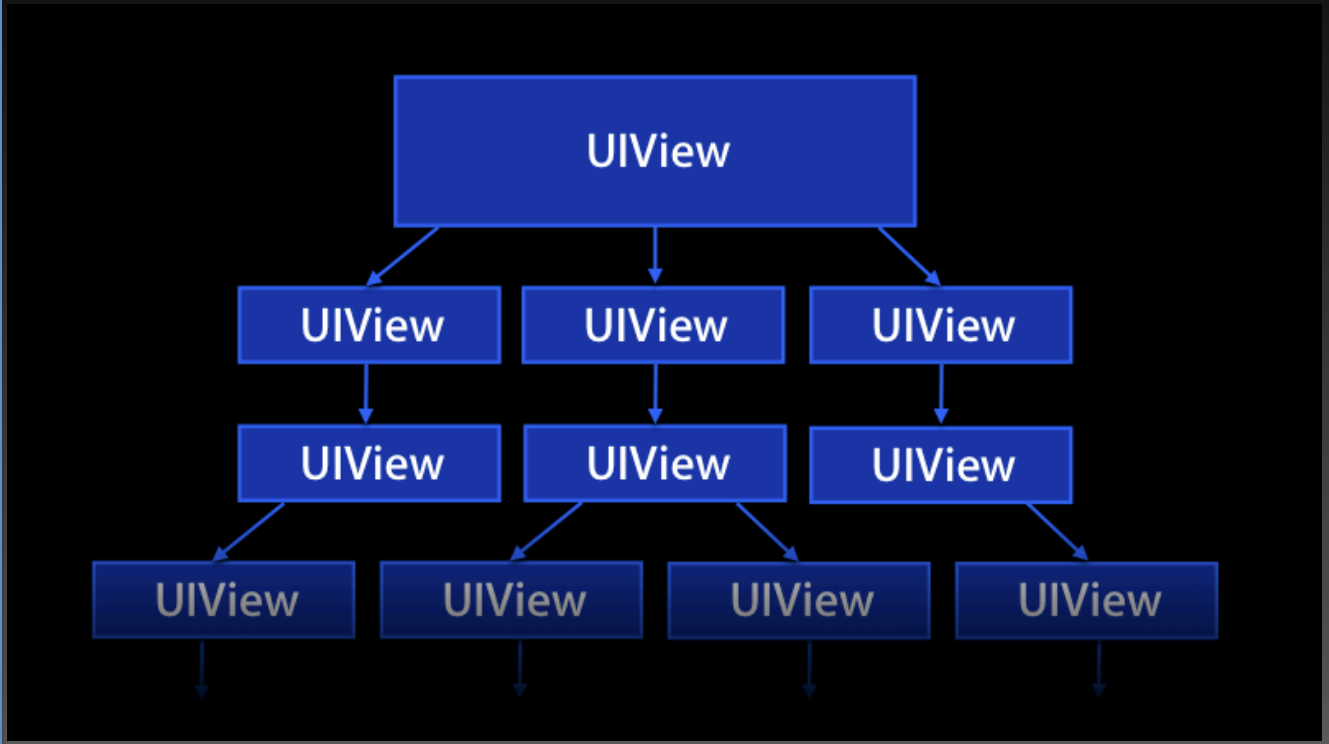
LoadMultipleViews

```
- (IBAction)loadAnotherView:(id)sender {
    if (!self.anotherView) {
        NSArray* nibArray = [[NSBundle mainBundle] loadNibNamed:@"anotherView" owner:self
options:nil];

        self.anotherView = [nibArray objectAtIndex:0];
    }
    self.mainView = self.view;
    self.view = self.anotherView;

}
- (IBAction)backToMainView:(id)sender {
    self.view = self.mainView;
}
```

Cách này không phải là cách khôn ngoan! Một UIViewController chỉ nên quản lý một main UIView

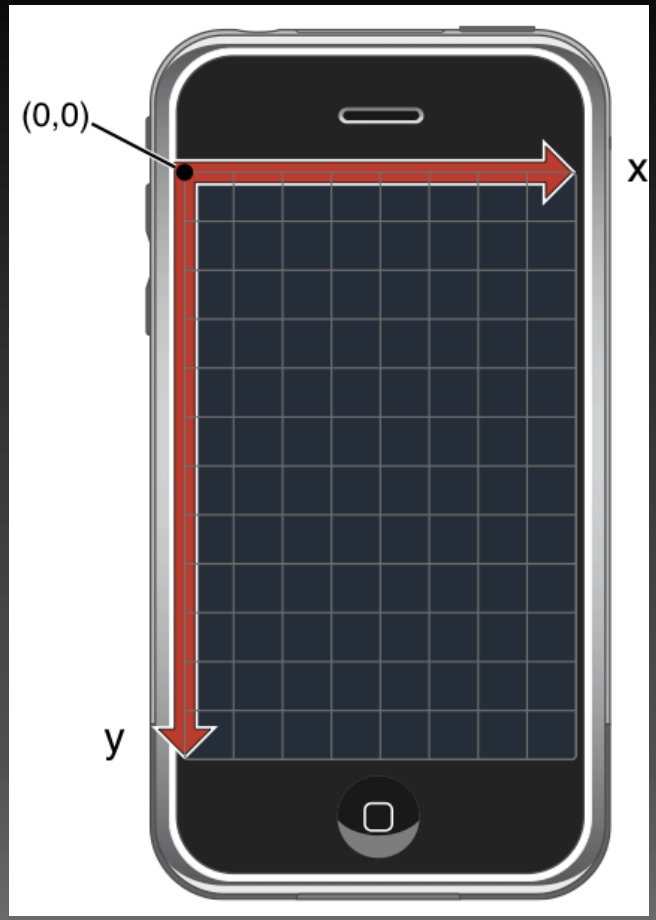
SuperView - SubViews

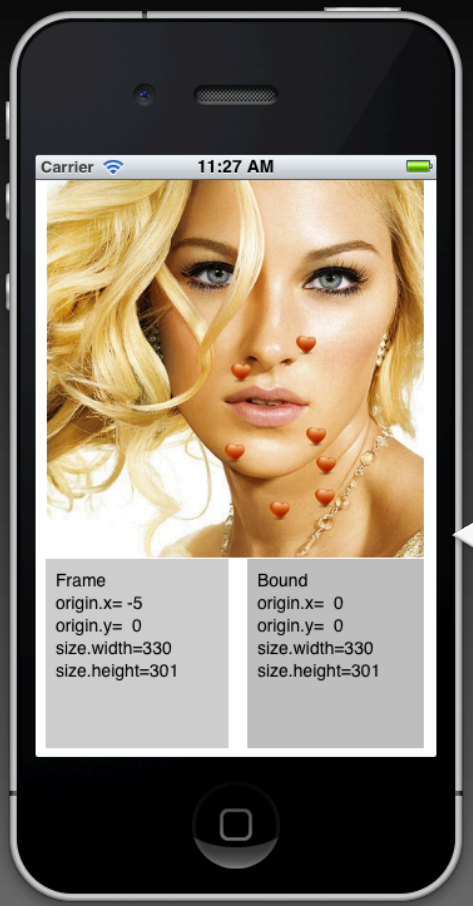


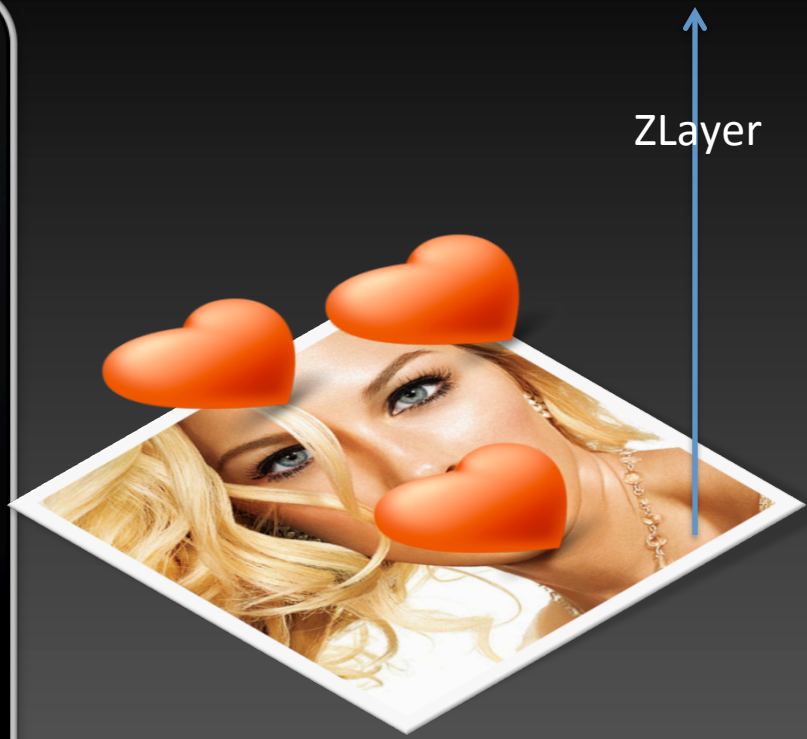


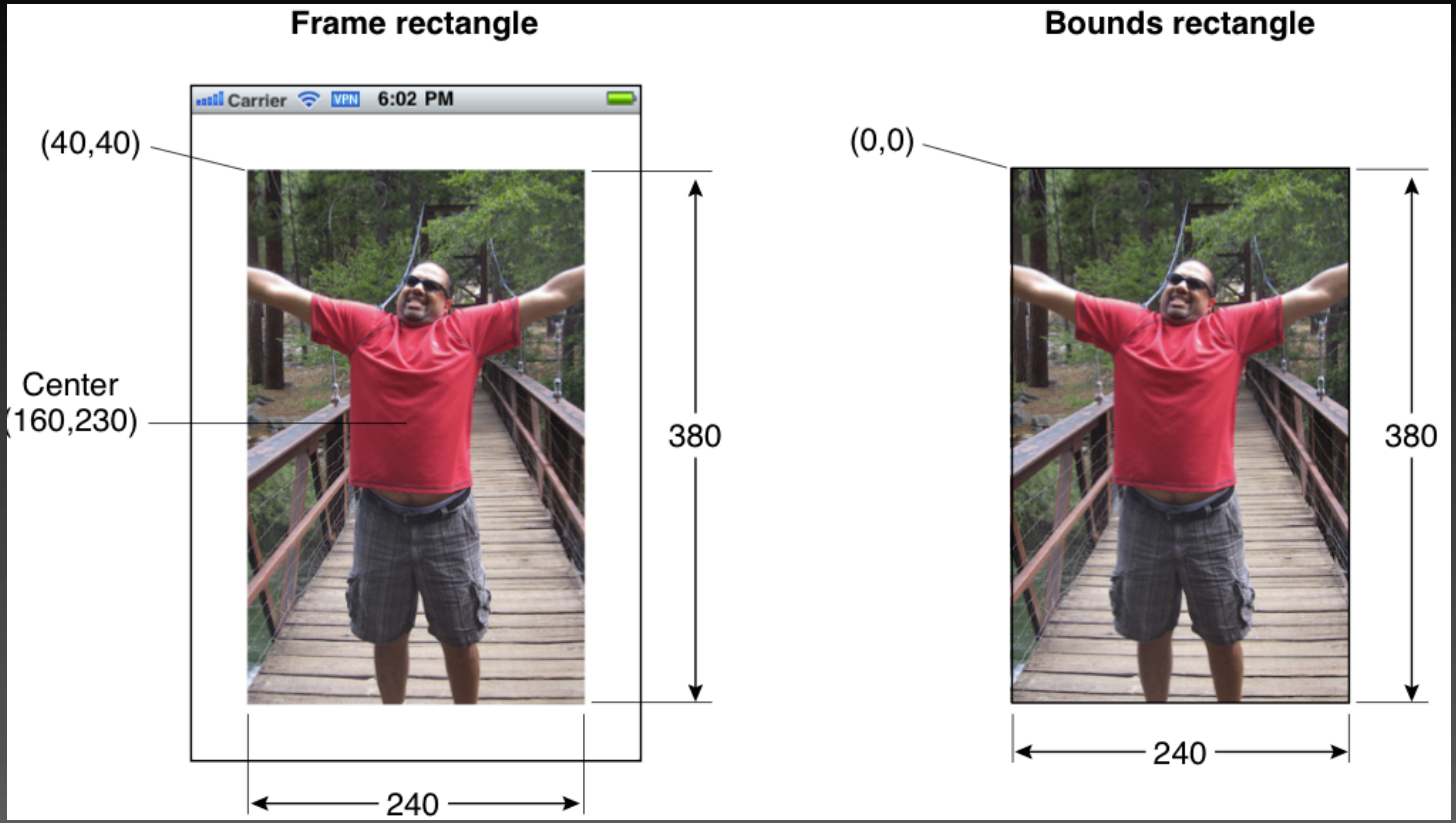
```objc
@interface UIView(UIViewHierarchy)

@property(nonatomic,readonly) UIView       *superview;
@property(nonatomic,readonly,copy) NSArray *subviews;
@property(nonatomic,readonly) UIWindow     *window;

- (void)removeFromSuperview;
- (void)insertSubview:(UIView *)view atIndex:(NSInteger)index;
- (void)exchangeSubviewAtIndex:(NSInteger)index1 withSubviewAtIndex:(NSInteger)index2;

- (void)addSubview:(UIView *)view;
- (void)insertSubview:(UIView *)view belowSubview:(UIView *)siblingSubview;
- (void)insertSubview:(UIView *)view aboveSubview:(UIView *)siblingSubview;

- (void)bringSubviewToFront:(UIView *)view;
- (void)sendSubviewToBack:(UIView *)view;

- (void)didAddSubview:(UIView *)subview;
- (void)willRemoveSubview:(UIView *)subview;

- (void)willMoveToSuperview:(UIView *)newSuperview;
- (void)didMoveToSuperview;
- (void)willMoveToWindow:(UIWindow *)newWindow;
- (void)didMoveToWindow;

- (BOOL)isDescendantOfView:(UIView *)view;  // returns YES for self.
- (UIView *)viewWithTag:(NSInteger)tag;     // recursive search. includes self

// Allows you to perform layout before the drawing cycle happens. -layoutIfNeeded forces layout early
- (void)setNeedsLayout;
- (void)layoutIfNeeded;
```

View - Geometry









```
@interface UIView(UIViewGeometry)

// animatable. do not use frame if view is transformed since it will not correctly reflect
@property(nonatomic) CGRect            frame;

// use bounds/center and not frame if non-identity transform. if bounds dimension is odd, c
@property(nonatomic) CGRect            bounds;      // default bounds is zero origin, frame
@property(nonatomic) CGPoint           center;      // center is center of frame. animatabl
@property(nonatomic) CGAffineTransform transform;   // default is CGAffineTransformIdentity
@property(nonatomic) CGFloat           contentScaleFactor __OSX_AVAILABLE_STARTING(__MAC_NA

@property(nonatomic,getter=isMultipleTouchEnabled) BOOL multipleTouchEnabled;   // default
@property(nonatomic,getter=isExclusiveTouch) BOOL       exclusiveTouch;         // default

- (UIView *)hitTest:(CGPoint)point withEvent:(UIEvent *)event;   // recursively calls -poin
- (BOOL)pointInside:(CGPoint)point withEvent:(UIEvent *)event;   // default returns YES if

- (CGPoint)convertPoint:(CGPoint)point toView:(UIView *)view;
- (CGPoint)convertPoint:(CGPoint)point fromView:(UIView *)view;
- (CGRect)convertRect:(CGRect)rect toView:(UIView *)view;
- (CGRect)convertRect:(CGRect)rect fromView:(UIView *)view;

@property(nonatomic) BOOL                autoresizesSubviews; // default is YES. if set, sub
@property(nonatomic) UIViewAutoresizing autoresizingMask;    // simple resize. default is U

- (CGSize)sizeThatFits:(CGSize)size;     // return 'best' size to fit given size. does not
```

Rendering

```
@interface UIView(UIViewRendering)

- (void)drawRect:(CGRect)rect;

- (void)setNeedsDisplay;
- (void)setNeedsDisplayInRect:(CGRect)rect;

@property(nonatomic)                  BOOL            clipsToBounds;             //
@property(nonatomic,copy)             UIColor         *backgroundColor;          //
@property(nonatomic)                  CGFloat         alpha;                     //
@property(nonatomic,getter=isOpaque)  BOOL            opaque;                    //
@property(nonatomic)                  BOOL            clearsContextBeforeDrawing; //
@property(nonatomic,getter=isHidden)  BOOL            hidden;                    //
@property(nonatomic)                  UIViewContentMode contentMode;             //
@property(nonatomic)                  CGRect          contentStretch __OSX_AVAILABLE

@end
```

Animation

```
@interface UIView(UIViewAnimation)

+ (void)beginAnimations:(NSString *)animationID context:(void *)context;  // addition
+ (void)commitAnimations;                                                 // starts u

// no getters. if called outside animation block, these setters have no effect.
+ (void)setAnimationDelegate:(id)delegate;                          // default = nil
+ (void)setAnimationWillStartSelector:(SEL)selector;                // default = NULL
+ (void)setAnimationDidStopSelector:(SEL)selector;                  // default = NULL
+ (void)setAnimationDuration:(NSTimeInterval)duration;              // default = 0.2
+ (void)setAnimationDelay:(NSTimeInterval)delay;                    // default = 0.0
+ (void)setAnimationStartDate:(NSDate *)startDate;                  // default = now
+ (void)setAnimationCurve:(UIViewAnimationCurve)curve;              // default = UIVi
+ (void)setAnimationRepeatCount:(float)repeatCount;                 // default = 0.0
+ (void)setAnimationRepeatAutoreverses:(BOOL)repeatAutoreverses;    // default = NO.
+ (void)setAnimationBeginsFromCurrentState:(BOOL)fromCurrentState;  // default = NO.

+ (void)setAnimationTransition:(UIViewAnimationTransition)transition forView:(UIView

+ (void)setAnimationsEnabled:(BOOL)enabled;                         // ignore any att
+ (BOOL)areAnimationsEnabled;

@end
```

Thuộc tính có thể animate

- frame—Use this to animate position and size changes for the view.
- bounds—Use this to animate changes to the size of the view.
- center—Use this to animate the position of the view.
- transform—Use this to rotate or scale the view.
- alpha—Use this to change the transparency of the view.
- backgroundColor—Use this to change the background color of the view.
- contentStretch—Use this to change how the view's contents stretch.

Animation with Block

```
@interface UIView(UIViewAnimationWithBlocks)

+ (void)animateWithDuration:(NSTimeInterval)duration delay:(NSTimeInterval)delay options:
(UIViewAnimationOptions)options animations:(void (^)(void))animations completion:(void (^)
(BOOL finished))completion __OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_4_0);

+ (void)animateWithDuration:(NSTimeInterval)duration animations:(void (^)(void))animations
completion:(void (^)(BOOL finished))completion
__OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_4_0); // delay = 0.0, options = 0

+ (void)animateWithDuration:(NSTimeInterval)duration animations:(void (^)(void))animations
__OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_4_0); // delay = 0.0, options = 0, completion =
NULL

+ (void)transitionWithView:(UIView *)view duration:(NSTimeInterval)duration options:
(UIViewAnimationOptions)options animations:(void (^)(void))animations completion:(void (^)
(BOOL finished))completion __OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_4_0);

+ (void)transitionFromView:(UIView *)fromView toView:(UIView *)toView duration:
(NSTimeInterval)duration options:(UIViewAnimationOptions)options completion:(void (^)(BOOL
finished))completion __OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_4_0); // toView added to
fromView.superview, fromView removed from its superview

@end
```

Gesture Recognition

- Trong tương lai sẽ học sâu hơn, hiện tại biết để dùng đã

```
@interface UIView (UIViewGestureRecognizers)

@property(nonatomic,copy) NSArray *gestureRecognizers
__OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_3_2);

- (void)addGestureRecognizer:
(UIGestureRecognizer*)gestureRecognizer
__OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_3_2);
- (void)removeGestureRecognizer:
(UIGestureRecognizer*)gestureRecognizer
__OSX_AVAILABLE_STARTING(__MAC_NA,__IPHONE_3_2);

@end
```

Các dẫn xuất của UIView

- UIAlertView, học luôn hôm nay
- UIActionSheet, học luôn
- UIWebView
- UIScrollView
- UITextView

Khi nào dùng Alert View, khi nào ActionSheet

Alerts, action sheets, and modal views are designed to communicate different things:

- Alerts give users important information that affects their use of the application (or the device). Alerts are usually unexpected, because they generally tell users about a problem or a change in the current situation that might require users to take action.

- Action sheets give users additional choices related to the action they are currently taking. Users learn to expect the appearance of an action sheet when they tap a toolbar button that begins either a potentially destructive action (such as deleting all recent calls) or an action that can be completed in different ways (such as a send action for which users can specify one of several destinations).

- Modal views provide more extensive functionality in the context of the current task or provide a way to perform a subtask directly related to the user's workflow.