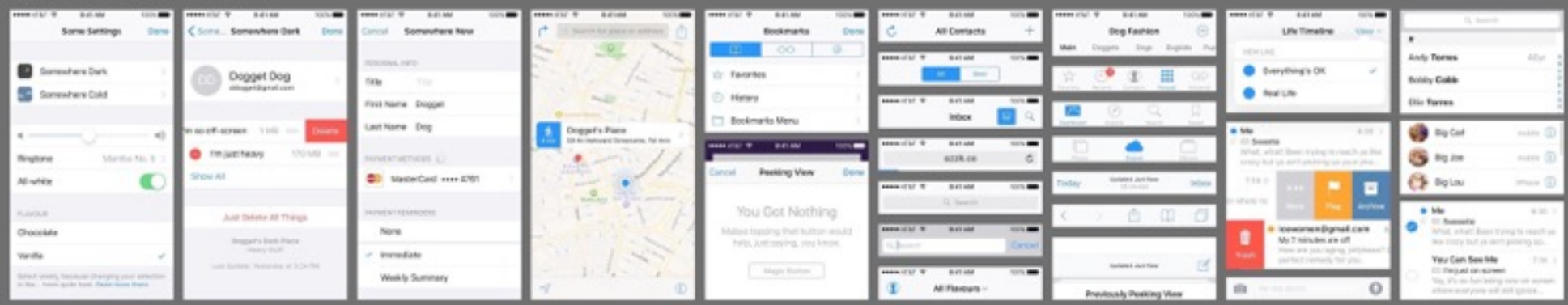
UIKit Framework

lecturer: Golubeva Maria

Every-Screen and Every-Center anyscreen.psd

Bars, views and lists blueprints.psd



Controls controls.psd

Dandy thingies dandy.psd

iOS 9 UI Kit

iPhone 5 (640 × 1136)

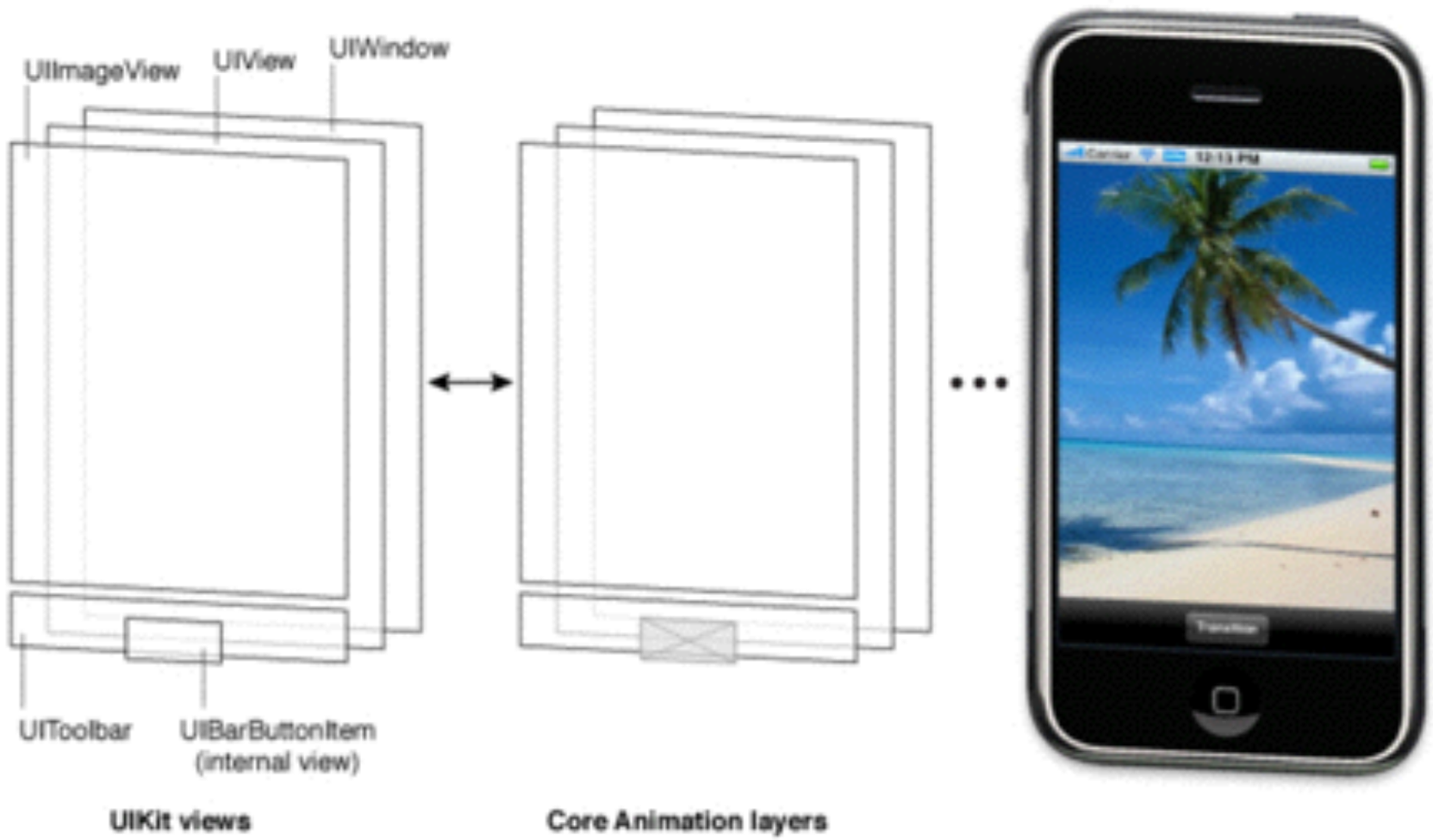
Made by Oz Pinhas. Get more of these goods at **http://ozzik.co** and **@ozzik**

Please don't go selling this PSD, claim it's yours or do bad stuff with it :)

Squircles are the fine work of Mark Edwards (bjango.com)

{{ random inspirational talk }}

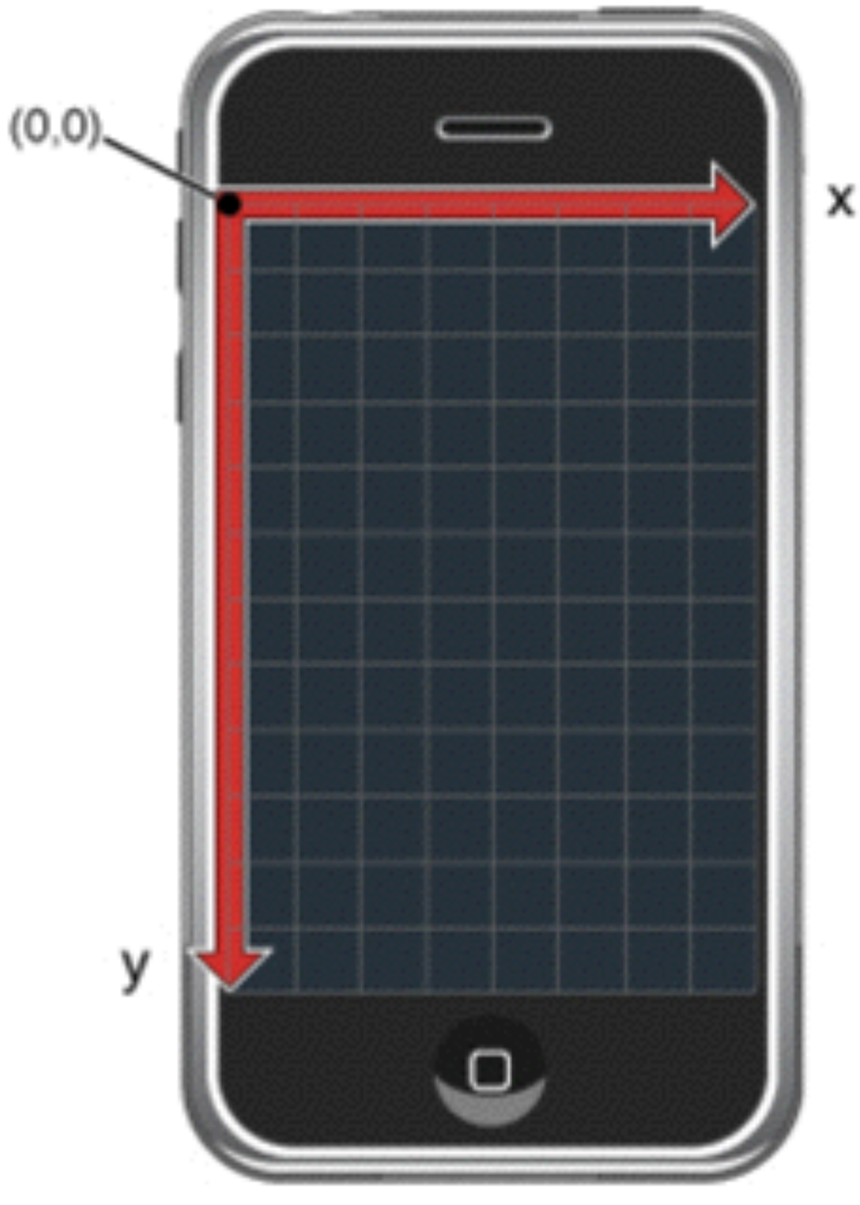
About Windows and Views



View and Window Architecture

View and Window Architecture

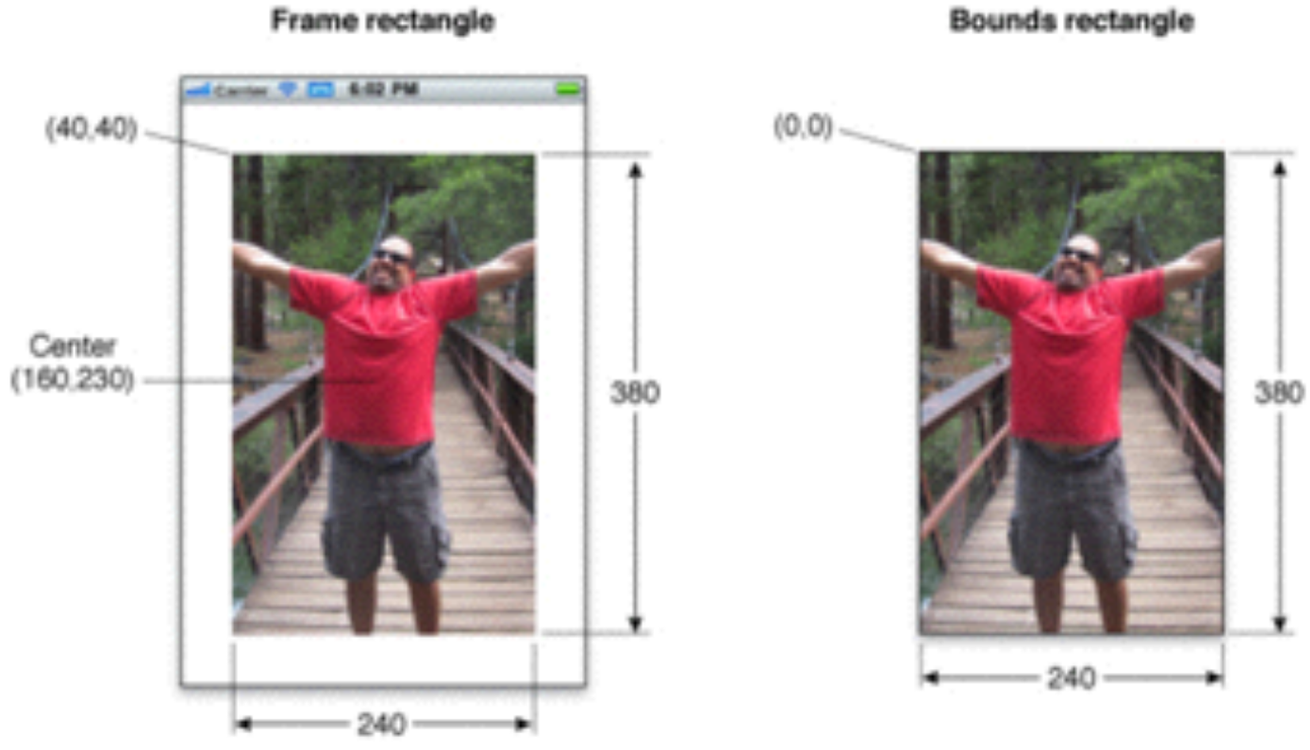
- **View Geometry and Coordinate Systems**



View and Window Architecture

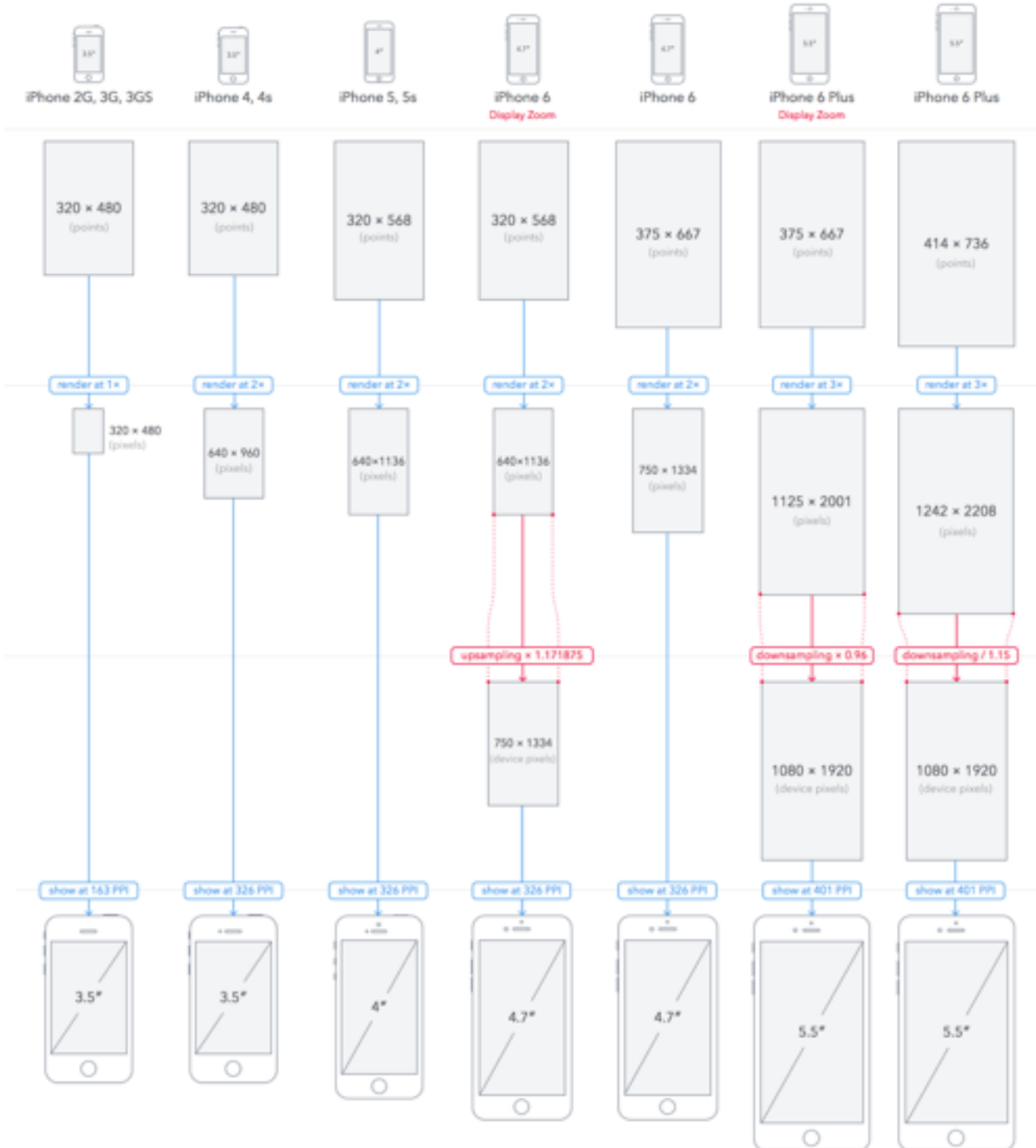
- View Geometry and Coordinate Systems

- **The Relationship of the Frame, Bounds, and Center Properties**



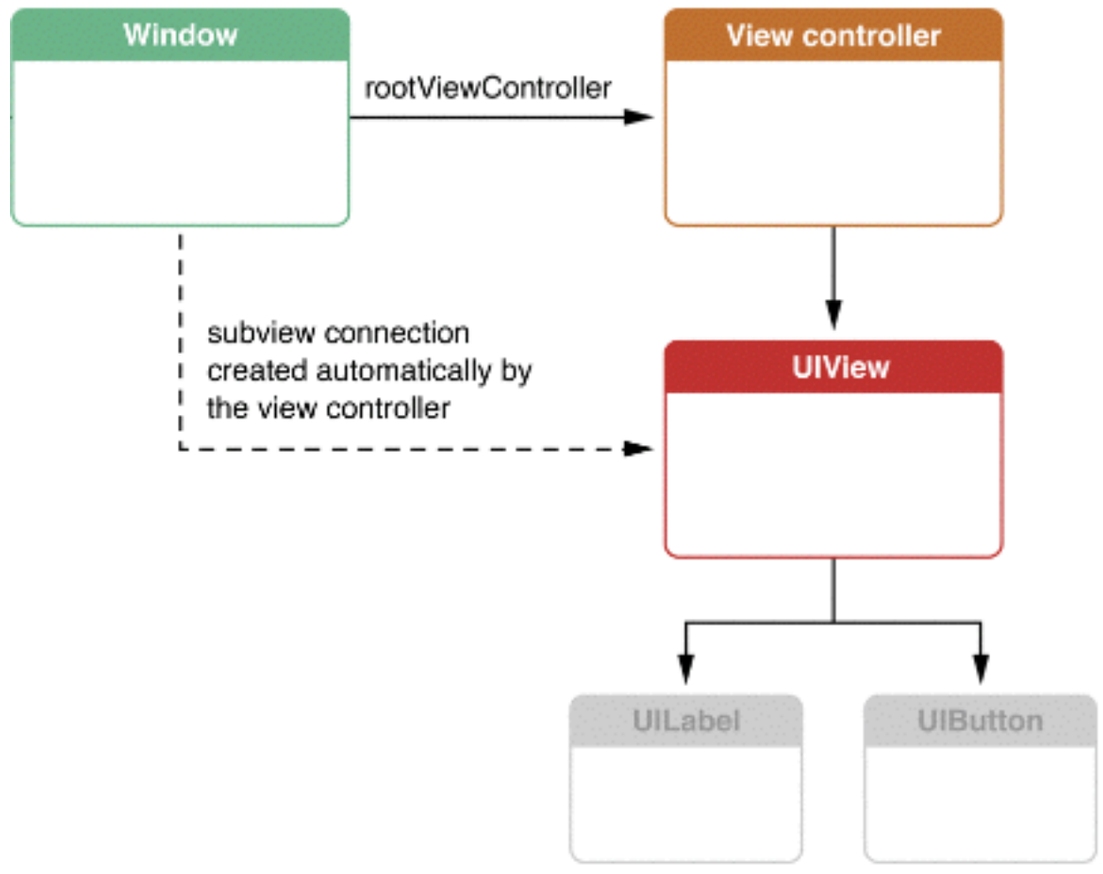
View and Window Architecture

- View Geometry and Coordinate Systems
- The Relationship of the Frame, Bounds, and Center Properties
- **Points vs Pixels**



UIWindow

- Contains application content
- Gives touches to UI objects
- Receives events about orientations changes.

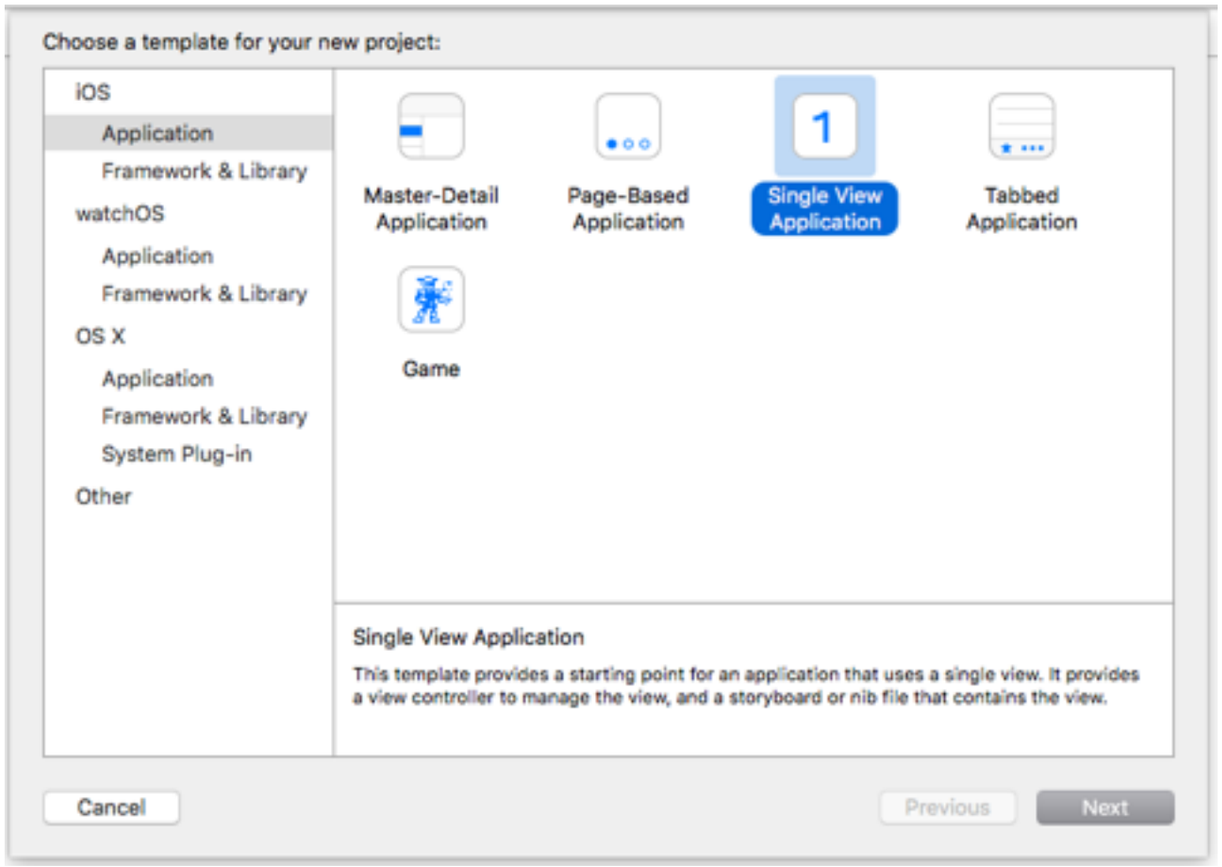


Creation of iOS project

- **Create new project**

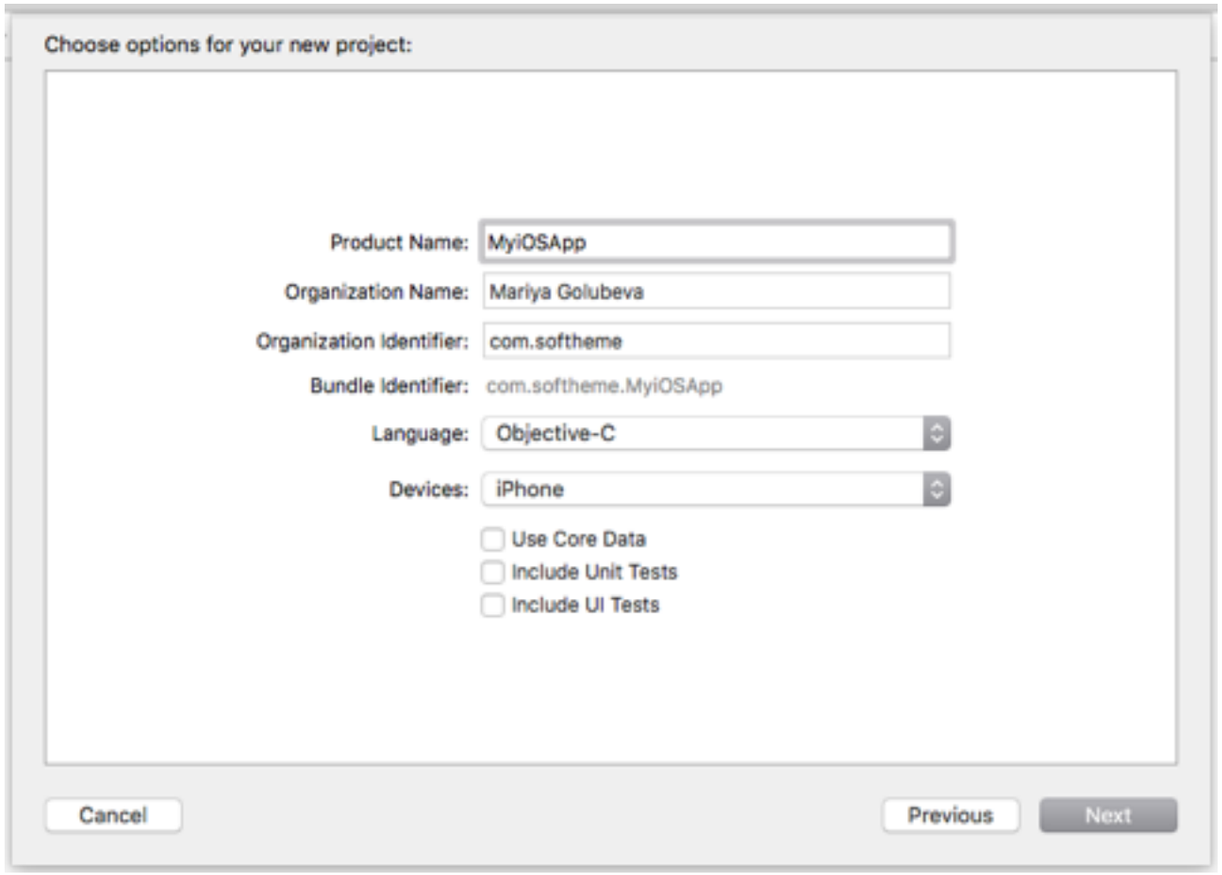
Creation of iOS project

- Create new project
- **Choose application type**



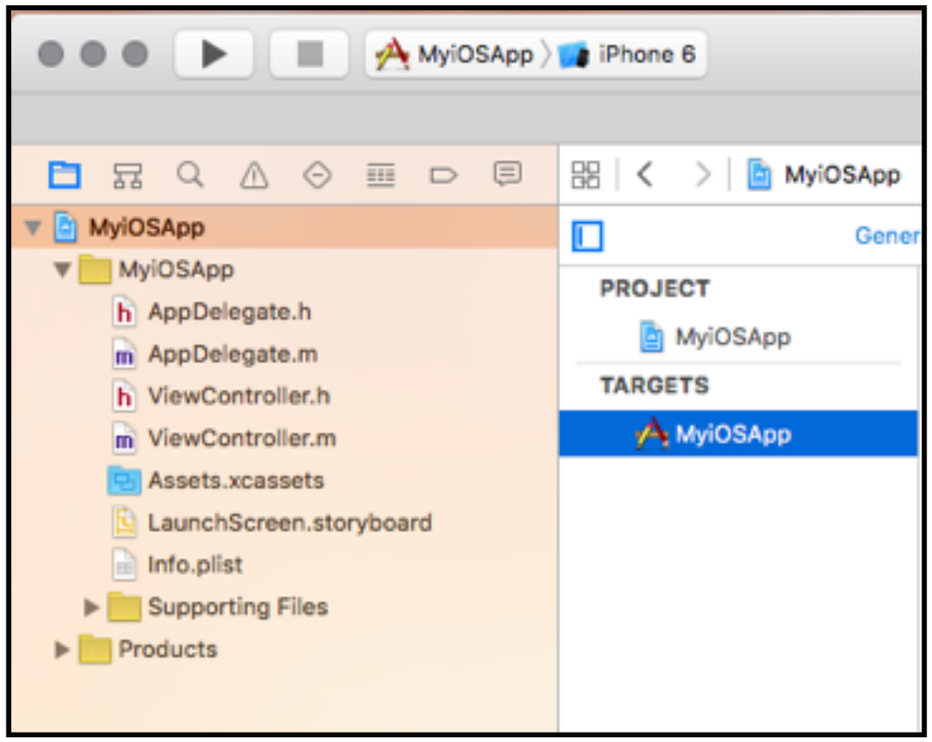
Creation of iOS project

- Create new project
- Choose application type
- **Enter the name for your project**



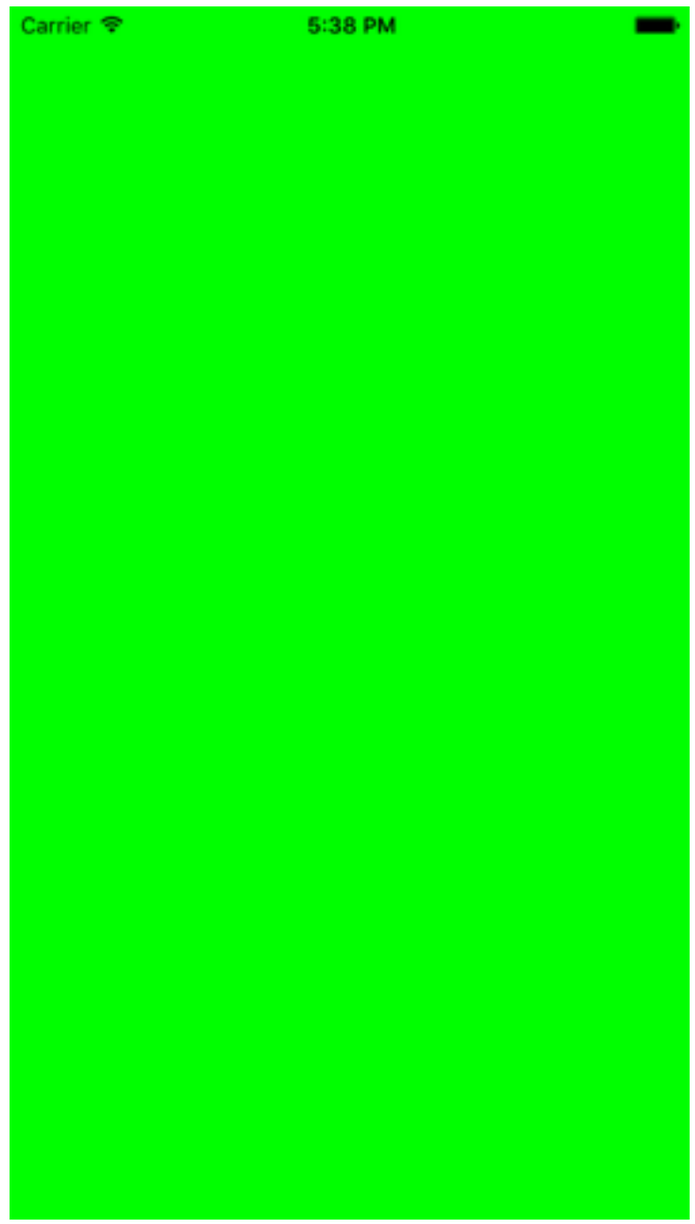
Creation of iOS project

- Create new project
- Choose application type
- Enter the name for your project
- **Delete *Main.storyboard* file, and in Target Settings remove “Main” in Main Interface.**



Create window programmatically

```
#import "AppDelegate.h"

@interface AppDelegate ()

@end

@implementation AppDelegate


- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{

    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]];
    self.window.backgroundColor = [UIColor greenColor];

    UIViewController *viewController = [[UIViewController
alloc] init];
    self.window.rootViewController = viewController;

    [self.window makeKeyAndVisible];


    return YES;
}

- (void)applicationWillResignActive:(UIApplication
*)application {
}
- (void)applicationDidEnterBackground:(UIApplication
*)application {
}
```



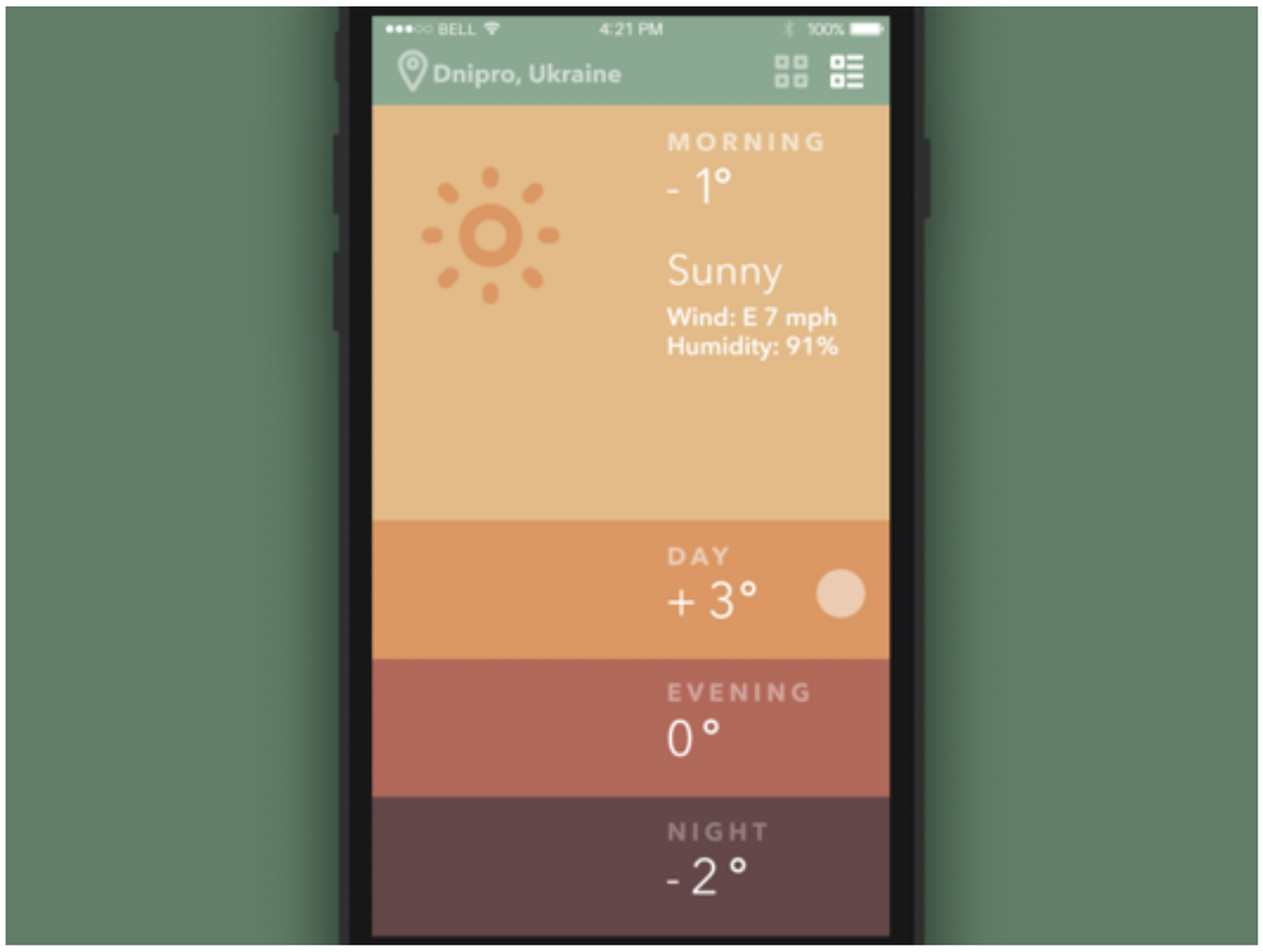
UIView

UIView

- **Layout and subview management**

UIView

- Layout and subview management
- **Drawing and animation**



UIView

- Layout and subview management
- Drawing and animation
- **Event handling**

Creation of UIView

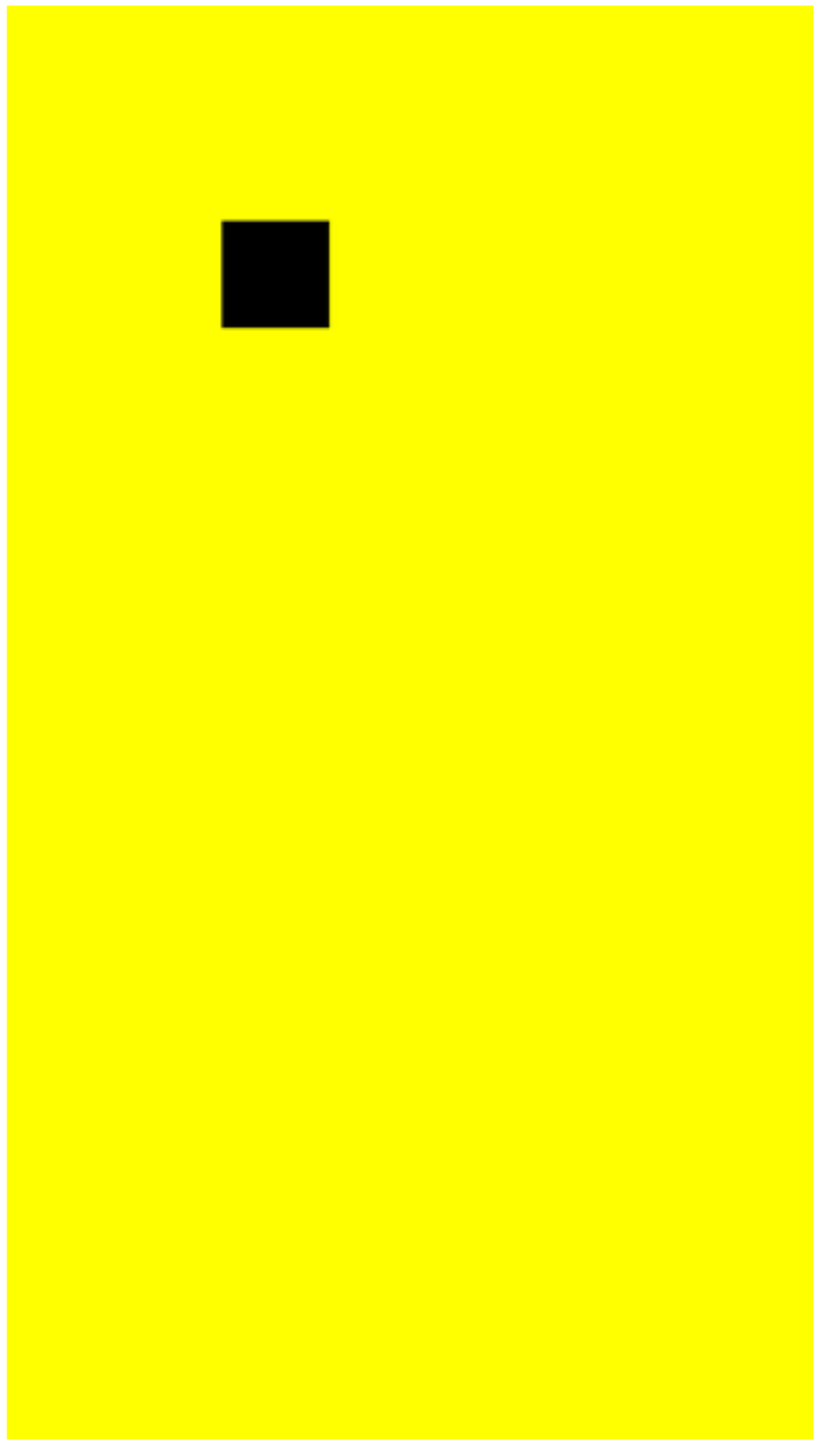
AppDelegate.m

```
#import "AppDelegate.h"
#import "ViewController.h"

@implementation AppDelegate


- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]];

    ViewController *viewController = [[ViewController alloc] init];
    self.window.rootViewController = viewController;

    [self.window makeKeyAndVisible];

    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application {
}
- (void)applicationDidEnterBackground:(UIApplication *)application
{
}
- (void)applicationWillEnterForeground:(UIApplication *)application
{
}
- (void)applicationDidBecomeActive:(UIApplication *)application {
}
- (void)applicationWillTerminate:(UIApplication *)application {
}

@end
```

ViewController.m

```
#import "ViewController.h"

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    self.view.backgroundColor = [UIColor
yellowColor];

    UIView *bg = [[UIView alloc]
initWithFrame:CGRectMake(100, 100, 50, 50)];
    bg.backgroundColor = [UIColor
blackColor];
    [self.view addSubview:bg];
}
- (BOOL)prefersStatusBarHidden {
    return YES;
}

@end
```

Creation of UIView

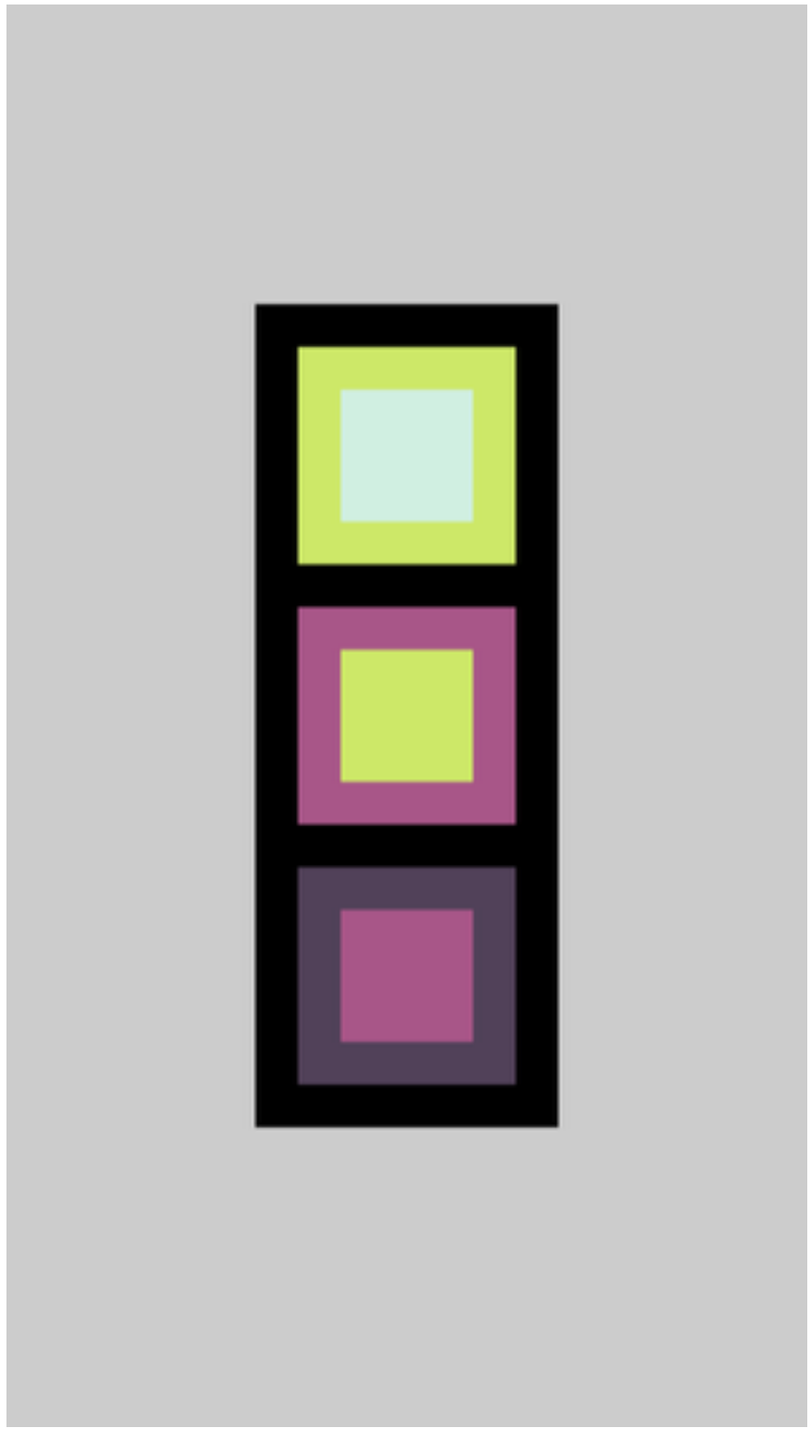


UIView hierarchy

ViewController.m

```
#import "ViewController.h"

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    self.view.backgroundColor = [UIColor colorWithRed:0.8f green:0.8f blue:0.8f alpha:1.f];

    NSArray *arrOfColors = @[[UIColor colorWithRed:81/255.f green:65/255.f blue:89/255.f alpha:1.f],
                             [UIColor colorWithRed:168/255.f green:86/255.f blue:136/255.f alpha:1.f],
                             [UIColor colorWithRed:208/255.f green:239/255.f blue:225/255.f alpha:1.f],
                             [UIColor colorWithRed:205/255.f green:232/255.f blue:104/255.f alpha:1.f],
                             [UIColor colorWithRed:140/255.f green:191/255.f blue:161/255.f alpha:1.f]
                             ];

    float offset = 20;
    float squareSize = roundf([UIScreen mainScreen].bounds.size.height/3-offset*6);
    float squareChildSize = squareSize-offset*2;
    float bgHeight = squareSize*3+offset*4;

    UIView *bg = [[UIView alloc] initWithFrame:CGRectMake(([UIScreen mainScreen].bounds.size.width-(squareSize+offset*2))/2,
                                                          ([UIScreen mainScreen].bounds.size.height-bgHeight)/2,
                                                          squareSize+offset*2,
                                                          bgHeight)];
    bg.backgroundColor = [UIColor blackColor];
    [self.view addSubview:bg];

    for (int i=0; i<3; i++) {
        int colorIndex1 = arc4random_uniform((int)arrOfColors.count);
        int colorIndex2 = colorIndex1;
        while (colorIndex2==colorIndex1) {
            colorIndex2 = arc4random_uniform((int)arrOfColors.count);
        }

        UIView *square = [[UIView alloc] initWithFrame:(CGRect){offset, offset+i*(offset+squareSize), squareSize, squareSize}];
        square.backgroundColor = arrOfColors[colorIndex1];
        [bg addSubview:square];

        UIView *squareChild = [[UIView alloc] initWithFrame:(CGRect){offset, offset, squareChildSize, squareChildSize}];
        squareChild.backgroundColor = arrOfColors[colorIndex2];
        [square addSubview:squareChild];
    }

}
- (BOOL)prefersStatusBarHidden {
    return YES;
}

@end
```

UIView hierarchy

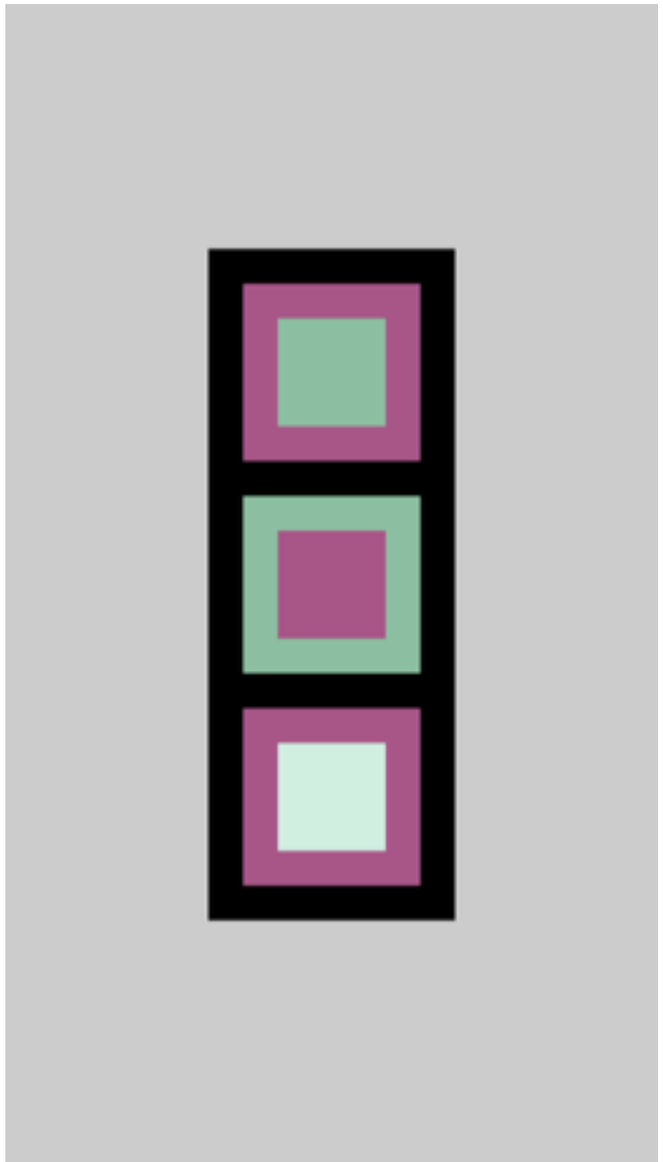


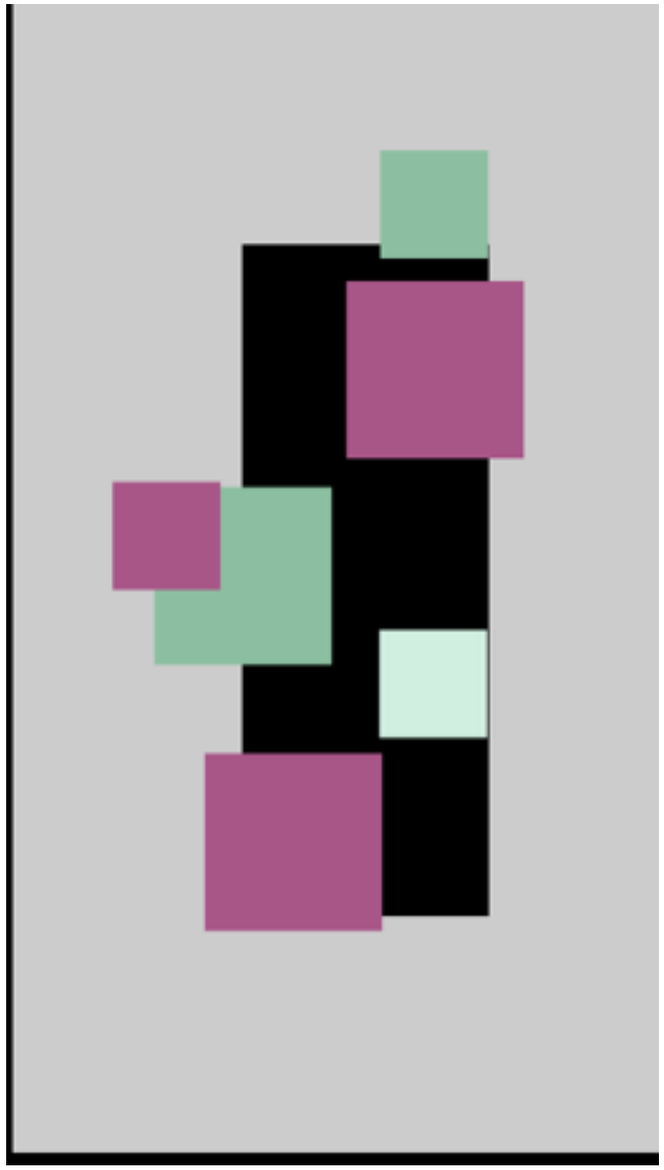
Touches handling

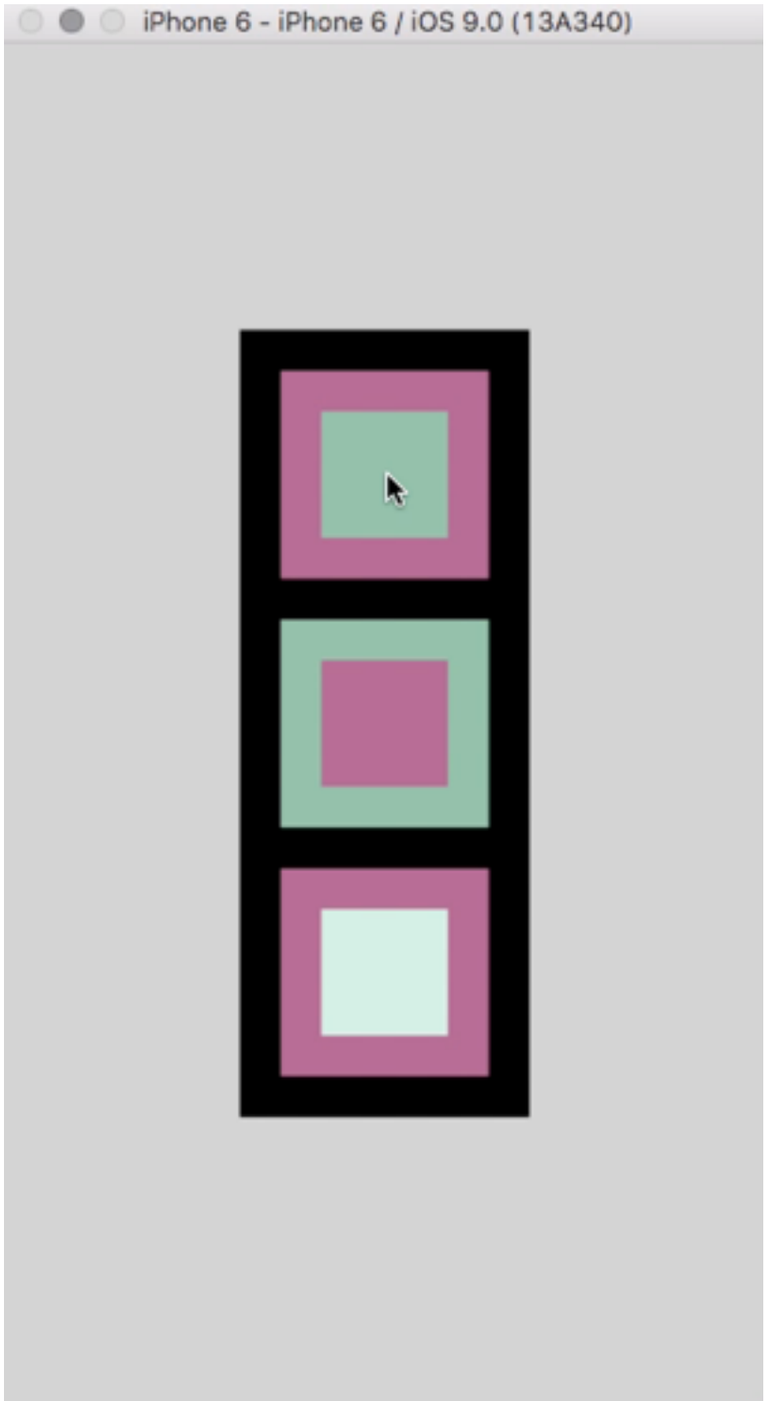
ViewController.m

```
#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)moveWithTouch:(UITouch*)touch {
    CGPoint position = [touch locationInView:touch.view];
    CGPoint previousPosition = [touch previousLocationInView:touch.view];

    CGRect frame = touch.view.frame;
    frame.origin.x -= (previousPosition.x-position.x);
    frame.origin.y -= (previousPosition.y-position.y);
    touch.view.frame = frame;
}

- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}
- (void)touchesMoved:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}
- (void)touchesEnded:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}
- (void)touchesCancelled:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}
@end
```

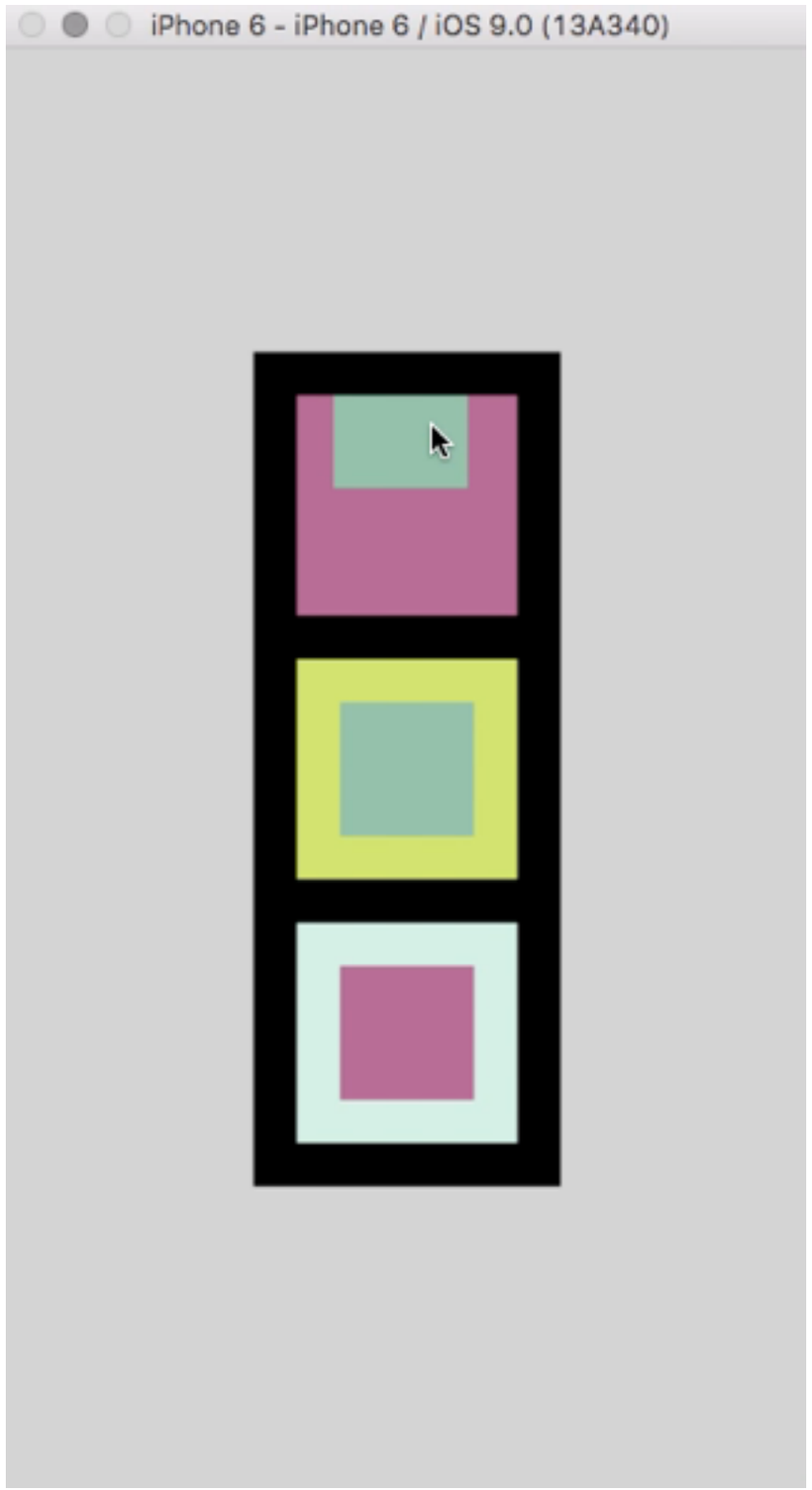
Touches handling





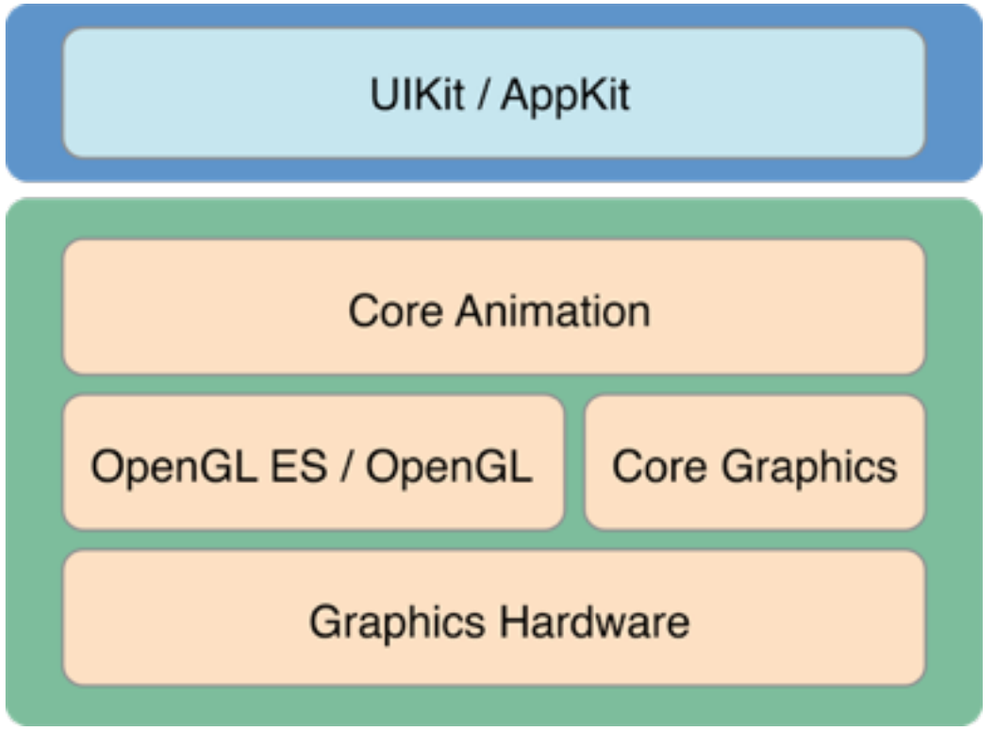


view.clipsToBounds = YES;



CALayer

- **It is Core Animation layer;**



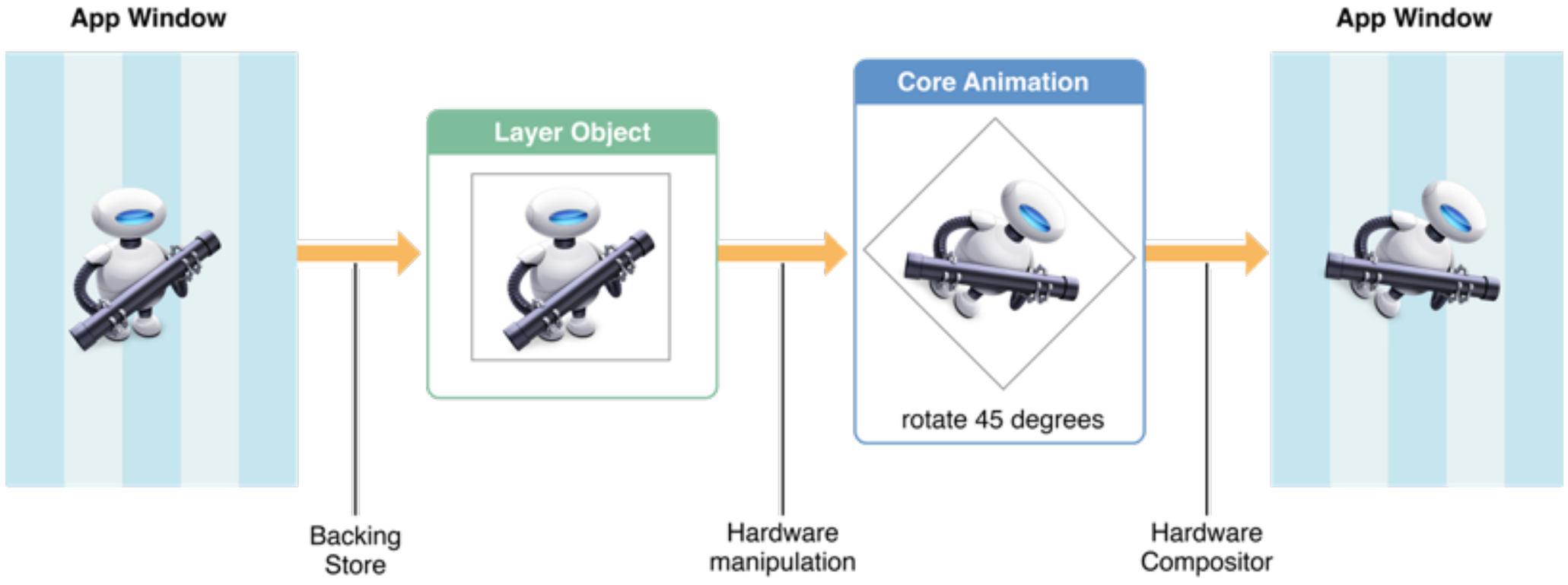
CALayer

- Core Animation layer;
- **The layer object for the view is stored in the view's *layer* property;**

```
self.view.layer;
```

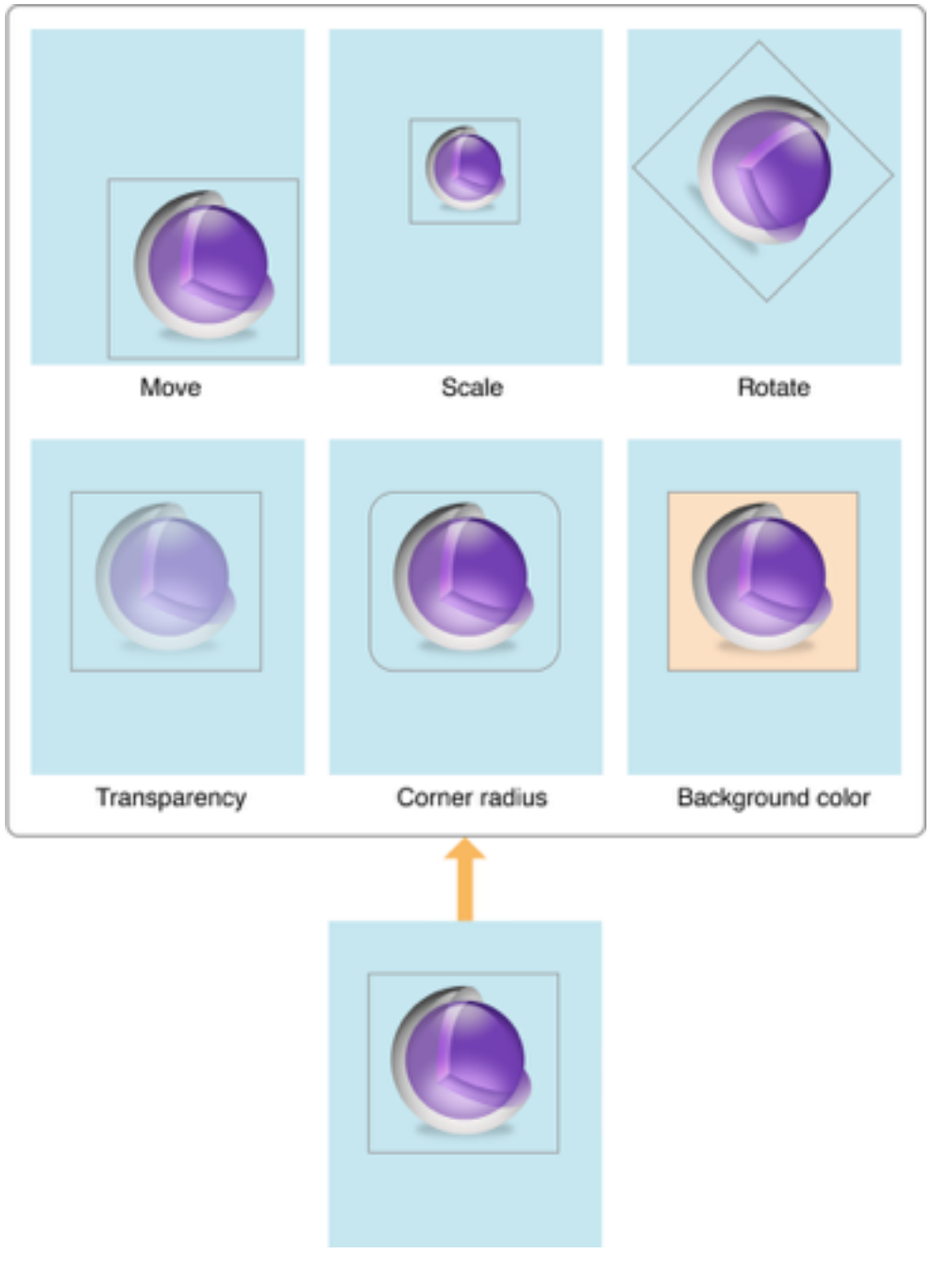
CALayer

- Core Animation layer;
- The layer object for the view is stored in the view's **layer** property;
- **Used to provide the backing store for views, for visual customisation;**



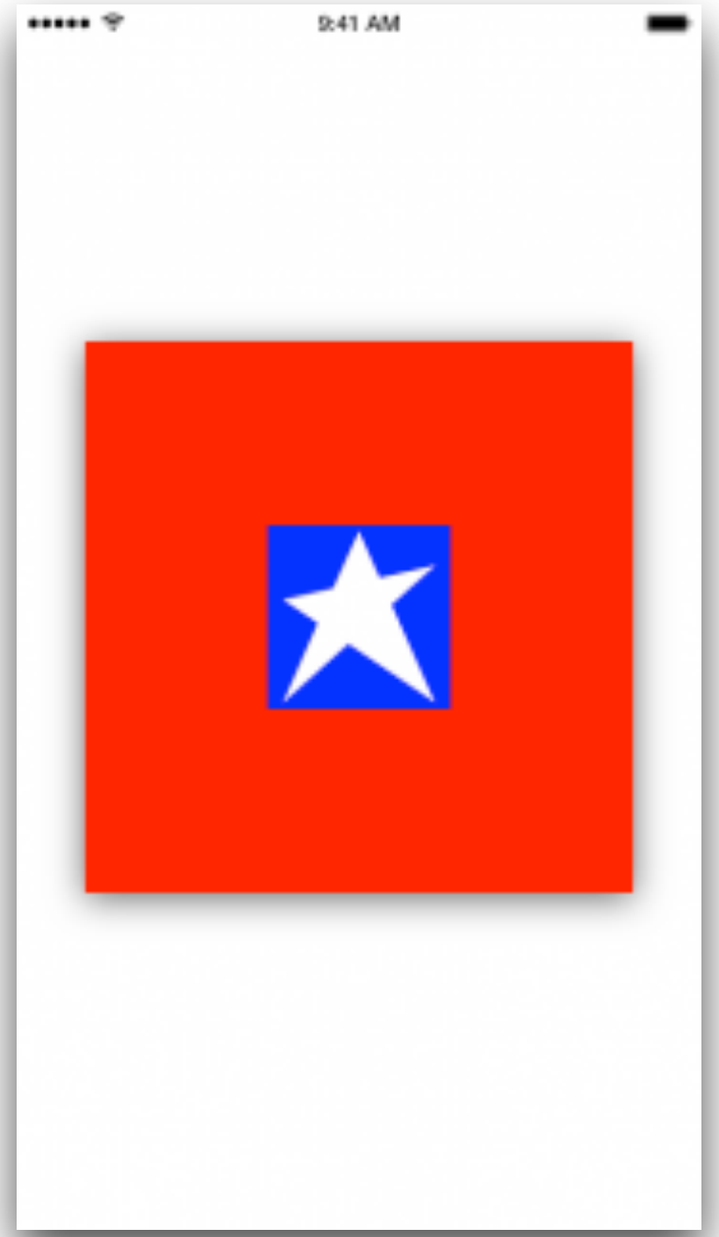
CALayer

- Core Animation layer;
- The layer object for the view is stored in the view's **layer** property;
- Used to provide the backing store for views, for visual customisation;
- **Allows you to perform animations on that content;**



CALayer

- Core Animation layer;
- The layer object for the view is stored in the view's **layer** property;
- Used to provide the backing store for views, for visual customisation;
- Allows you to perform animations on that content;
- **Also can be used without a view to display content;**



Animations

What can be animated?

Two ways for animation creation

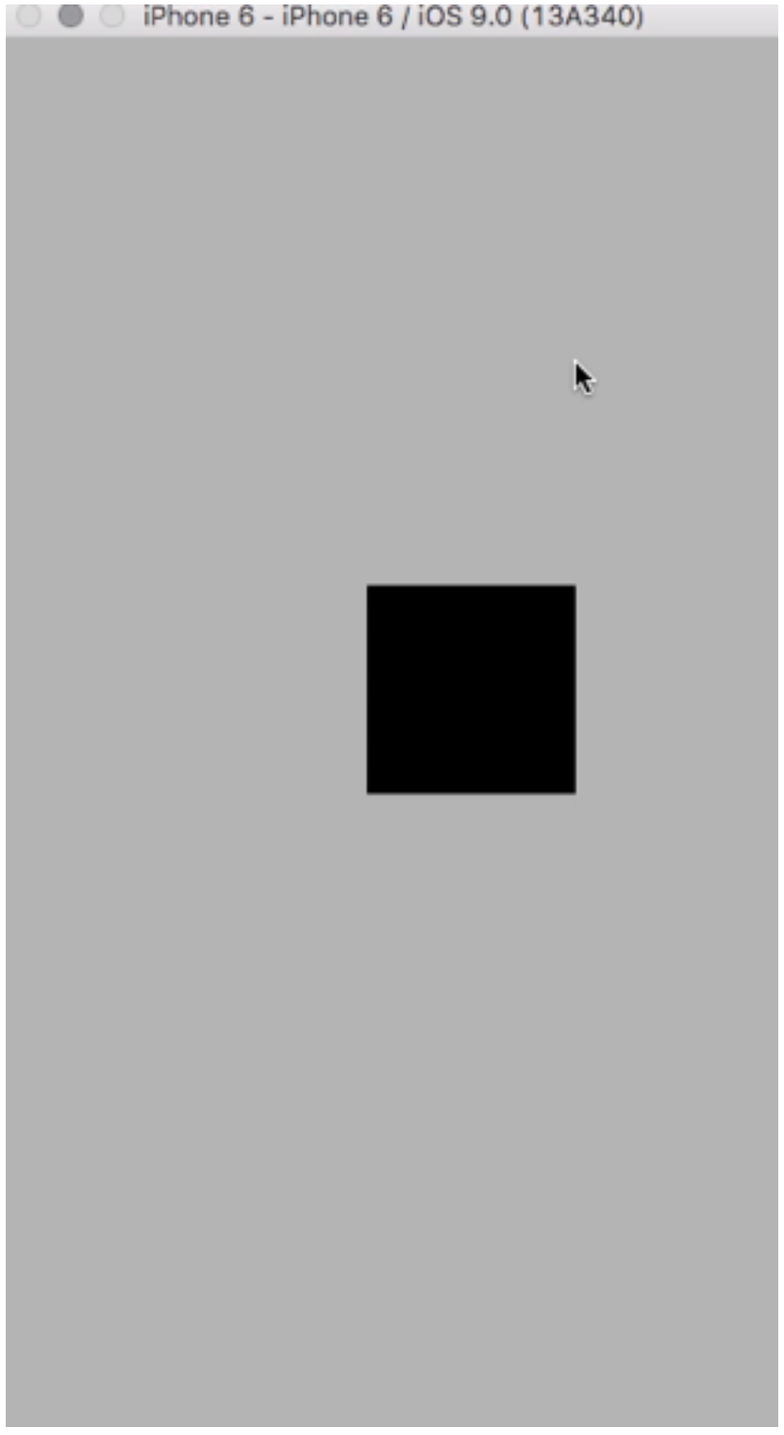
Using the Block-Based Methods

```
#import "ViewController.h"

@interface ViewController ()
@property (nonatomic, weak) UIView *movedView;
@end

@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor
whiteColor];
    UIView *v = [[UIView alloc] initWithFrame:
(CGRect){0, 0, 100, 100}];
    v.backgroundColor = [UIColor blackColor];
    [self.view addSubview:v];
    self.movedView = v;
}
- (BOOL)prefersStatusBarHidden {
    return YES;
}
- (void)moveWithTouch:(UITouch*)touch {
    CGPoint position = [touch
                        locationInView:touch.view];

    [UIView animateWithDuration:0.3f animations:^{
        self.movedView.center = position;
    }];
}
- (void)touchesBegan:(NSSet<UITouch *> *)touches
           withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}

@end
```

Using the Begin/Commit Methods

```
#import "ViewController.h"

@interface ViewController ()
@property (nonatomic, weak) UIView *movedView;
@end

@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor whiteColor];
    UIView *v = [[UIView alloc] initWithFrame:
(CGRect){0, 0, 100, 100}];
    v.backgroundColor = [UIColor blackColor];
    [self.view addSubview:v];
    self.movedView = v;
}
- (BOOL)prefersStatusBarHidden {
    return YES;
}
- (void)moveWithTouch:(UITouch*)touch {
    CGPoint position = [touch
                        locationInView:touch.view];

    [UIView beginAnimations:@"MyAnimation"
                                          context:nil];
    [UIView setAnimationDuration:0.3f];
    self.movedView.center = position;
    [UIView commitAnimations];
}
- (void)touchesBegan:(NSSet<UITouch *> *)touches
           withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}
@end
```

Two ways for animation creation

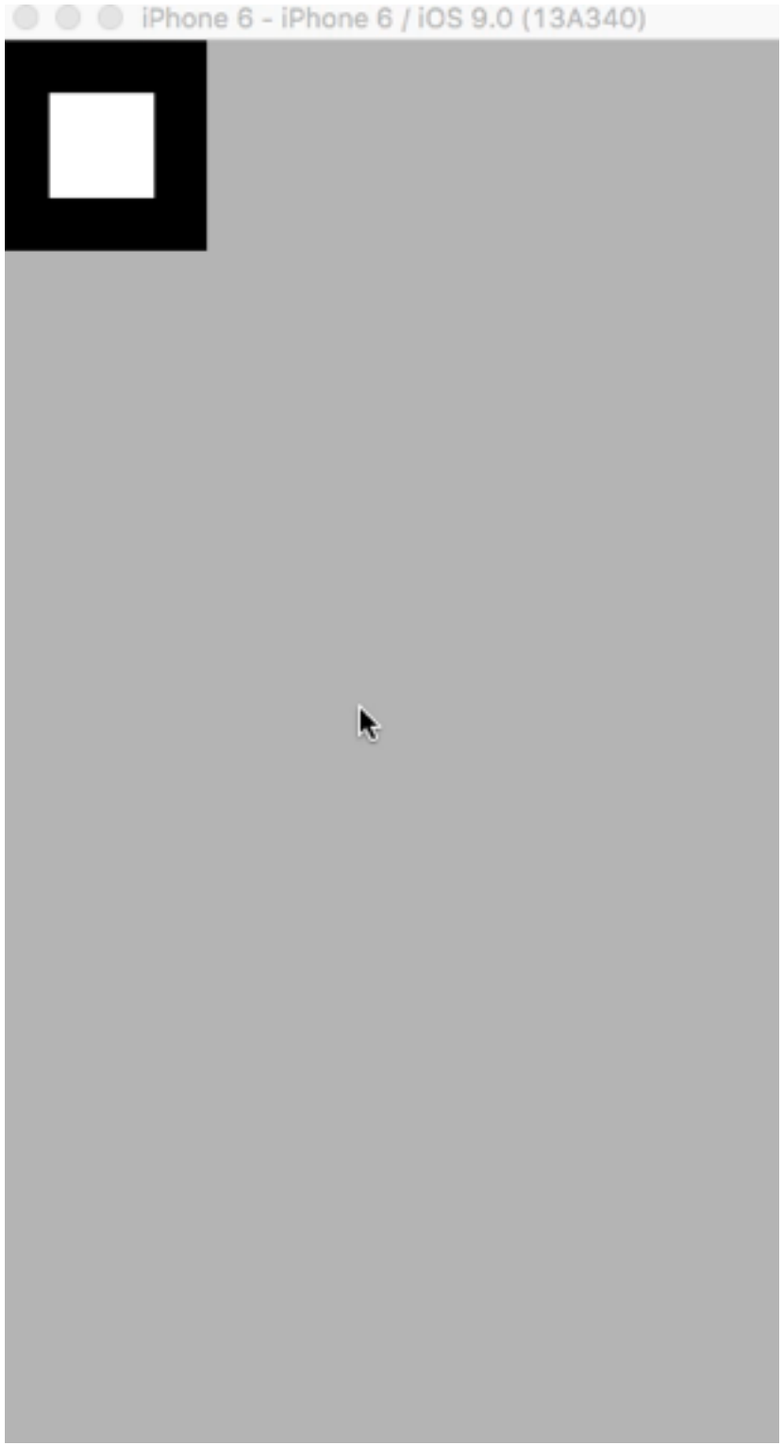


Animations

ViewController.m

```
#import "ViewController.h"
@interface ViewController ()
@property (nonatomic, weak) UIView *movedView;
@property (nonatomic, weak) UIView *movedView2;
@end
@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];

    self.view.backgroundColor = [UIColor whiteColor];

    UIView *v = [[UIView alloc] initWithFrame:(CGRect){0, 0, 100, 100}];
    v.backgroundColor = [UIColor blackColor];
    [self.view addSubview:v];
    self.movedView = v;

    UIView *v2 = [[UIView alloc] initWithFrame:(CGRect){25, 25, 50, 50}];
    v2.backgroundColor = [UIColor whiteColor];
    [self.movedView addSubview:v2];
    self.movedView2 = v2;
}
- (BOOL)prefersStatusBarHidden {
    return YES;
}
- (void)moveWithTouch:(UITouch*)touch {
    CGPoint position = [touch locationInView:touch.view];

    [UIView animateWithDuration:0.3f animations:^{
        self.movedView.center = position;
        self.movedView2.transform = CGAffineTransformMakeRotation(M_PI);
    } completion:^(BOOL finished) {
        [UIView animateWithDuration:0.3f animations:^{
            self.movedView2.transform = CGAffineTransformMakeRotation(0);
        }];
    }];
}
- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}
@end
```

Animations



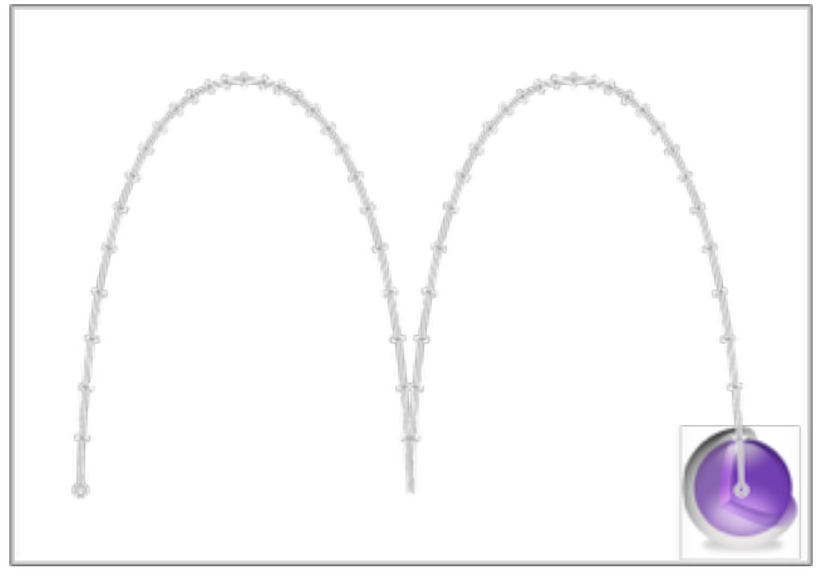
Configuring an Animation Delegate

ViewController.m

```
#import "ViewController.h"

@interface ViewController ()
@property (nonatomic, weak) UIView *movedView;
@property (nonatomic, weak) UIView *movedView2;
@end
@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    self.view.backgroundColor = [UIColor lightGrayColor];

    UIView *v = [[UIView alloc] initWithFrame:(CGRect){0, 0, 100, 100}];
    v.backgroundColor = [UIColor blackColor];
    [self.view addSubview:v];
    self.movedView = v;

    UIView *v2 = [[UIView alloc] initWithFrame:(CGRect){25, 25, 50, 50}];
    v2.backgroundColor = [UIColor whiteColor];
    [self.movedView addSubview:v2];
    self.movedView2 = v2;

      [UIView setAnimationDelegate:self];
}
- (BOOL)prefersStatusBarHidden {
    return YES;
}
- (void)moveWithTouch:(UITouch*)touch {
    CGPoint position = [touch locationInView:touch.view];

    [UIView animateWithDuration:0.3f animations:^{
        self.movedView.center = position;
        self.movedView2.transform = CGAffineTransformMakeRotation(M_PI);
    } completion:^(BOOL finished) {
        [UIView animateWithDuration:0.3f animations:^{
            self.movedView2.transform = CGAffineTransformMakeRotation(0);
        }];
    }];
}
- (void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event {
    UITouch *touch = [touches anyObject];
    [self moveWithTouch:touch];
}
- (void)animationWillStart:(NSString *)animationID context:(void *)context {
    NSLog(@"Animation Started!");
}
- (void)animationDidStop:(NSString *)animationID finished:(NSNumber *)finished context:(void *)context {
    NSLog(@"Animation Ended!");
}
@end
```

Animation of CALayer

```
CGMutablePathRef thePath = CGPathCreateMutable();

CGPathMoveToPoint(thePath,NULL,74.0,74.0);

CGPathAddCurveToPoint(thePath, NULL,  74.0, 500.0,
                                     320.0, 500.0,
                                     320.0, 74.0);

CGPathAddCurveToPoint(thePath, NULL, 320.0, 500.0,
                                     566.0, 500.0,
                                     566.0, 74.0);

CAKeyframeAnimation * theAnimation;

theAnimation = [CAKeyframeAnimation
                    animationWithKeyPath:@"position"];
theAnimation.path = thePath;
theAnimation.duration=5.0;

[theLayer addAnimation:theAnimation
                forKey:@"position"];
```



Animation options

Animation options

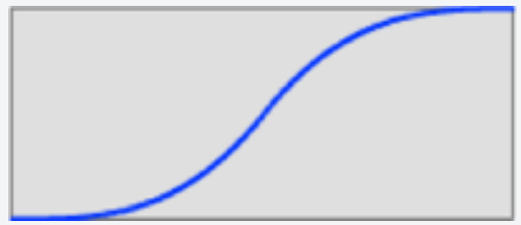
- **Where they are?**

```
[UIView animateWithDuration:0.5f
                      delay:0.f
                    options:UIViewAnimationOptionCurveLinear|
                            UIViewAnimationOptionRepeat|
                            UIViewAnimationOptionAutoreverse
                 animations:^{
                       self.player2_Ship.alpha = 0.2f;
                 }
                 completion:nil];
```

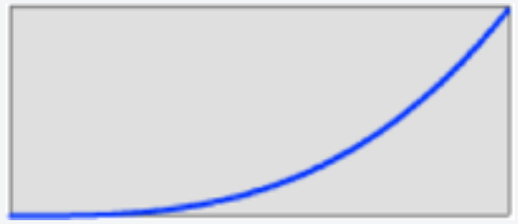
Animation options

- Where they are?
- **Curve Animation options;**

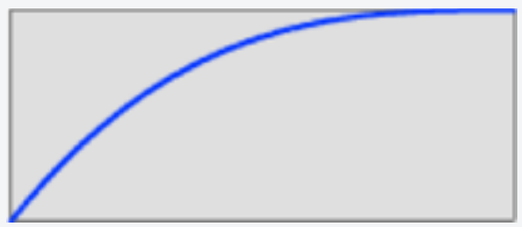
`UIViewAnimationOptionCurveEaseInOut`



`UIViewAnimationOptionCurveEaseIn`



`UIViewAnimationOptionCurveEaseOut`



`UIViewAnimationOptionCurveLinear`

Animation options

- Where they are?
- Curve Animation options;
- **Transitions Animation Options;**

```
UIViewAnimationOptionTransitionNone  // default
UIViewAnimationOptionTransitionFlipFromLeft
UIViewAnimationOptionTransitionFlipFromRight
UIViewAnimationOptionTransitionCurlUp
UIViewAnimationOptionTransitionCurlDown
UIViewAnimationOptionTransitionCrossDissolve
UIViewAnimationOptionTransitionFlipFromTop
UIViewAnimationOptionTransitionFlipFromBottom
```

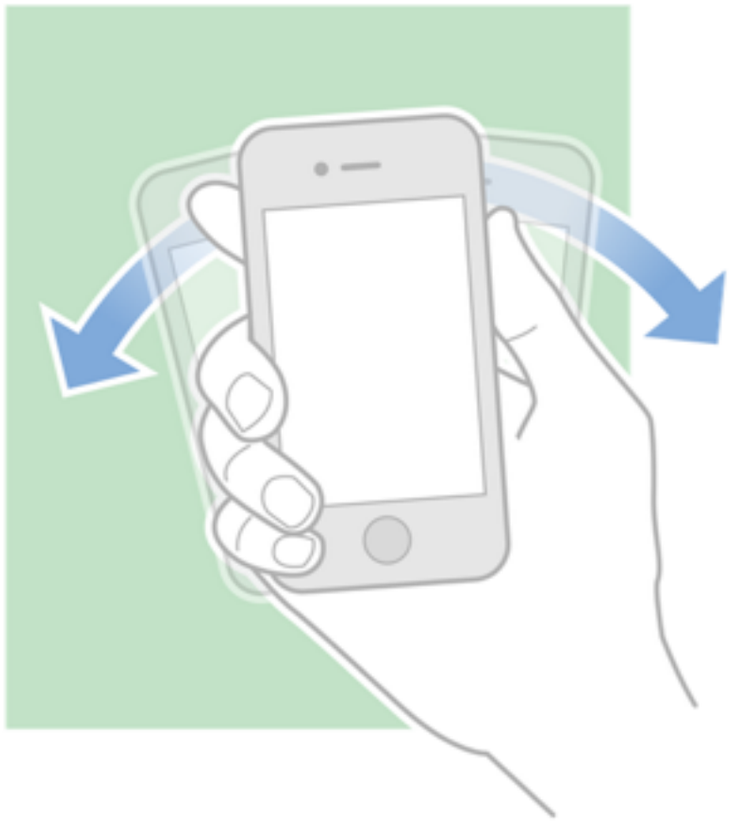
Animation options

- Where they are?
- Curve Animation options;
- Transitions Animation Options;
- **Other Animation Options**

```
UIViewAnimationOptionLayoutSubviews

//turn on user interaction while animating
UIViewAnimationOptionAllowUserInteraction

// start all views from current value, not initial value
UIViewAnimationOptionBeginFromCurrentState

// repeat animation indefinitely
UIViewAnimationOptionRepeat

// if repeat, run animation back and forth
UIViewAnimationOptionAutoreverse

// ignore nested duration
UIViewAnimationOptionOverrideInheritedDuration

// ignore nested curve
UIViewAnimationOptionOverrideInheritedCurve

// animate contents (applies to transitions only)
UIViewAnimationOptionAllowAnimatedContent

// flip to/from hidden state instead of adding/removing
UIViewAnimationOptionShowHideTransitionViews

UIViewAnimationOptionOverrideInheritedOptions
```

About Events in iOS

Multitouch events



Accelerometer events

Remote control events

Gesture Recognizers

Gesture Recognizers

- **Tapping (any number of taps)**

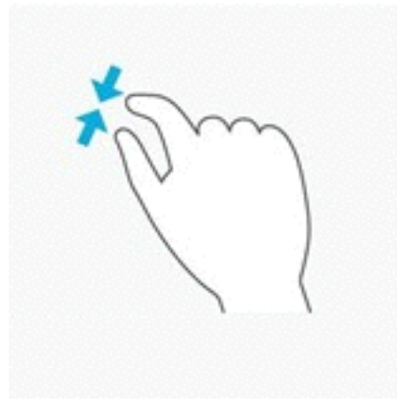
UITapGestureRecognizer

```
UITapGestureRecognizer *tap = [[UITapGestureRecognizer
alloc] initWithTarget:self action:@selector(onTap)];

tap.numberOfTapsRequired = 2;
tap.numberOfTouchesRequired = 4;
[self.view addGestureRecognizer:tap];
```

Gesture Recognizers

- Tapping (any number of taps)
- **Pinching in and out (for zooming a view)**

UIPinchGestureRecognizer

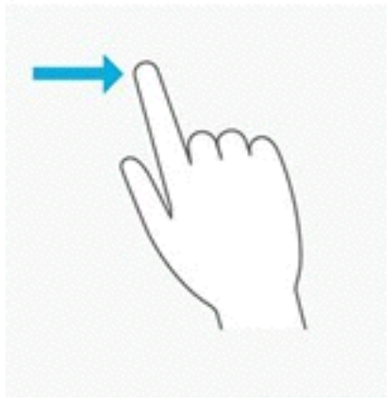


```
UIPinchGestureRecognizer *pinch =
[[UIPinchGestureRecognizer alloc] initWithTarget:self
                                          action:@selector(onPinch:)];

[self.view addGestureRecognizer:pinch];
```

Gesture Recognizers

- Tapping (any number of taps)
- Pinching in and out (for zooming a view)
- **Panning or dragging**

UIPanGestureRecognizer

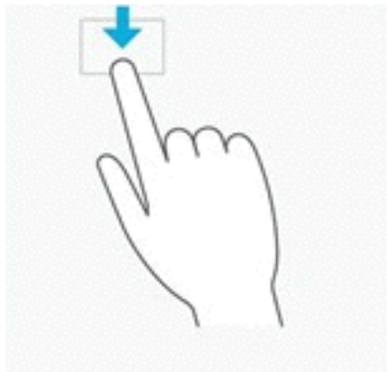


```
UIPanGestureRecognizer *pan = [[UIPanGestureRecognizer
alloc] initWithTarget:self action:@selector(onPan:)];

pan.maximumNumberOfTouches = 2;
pan.minimumNumberOfTouches = 1;
[self.view addGestureRecognizer:pan];
```

Gesture Recognizers

- Tapping (any number of taps)
- Pinching in and out (for zooming a view)
- Panning or dragging
- **Swiping (in any direction)**

UISwipeGestureRecognizer



```
UISwipeGestureRecognizer *swipe =
[[UISwipeGestureRecognizer alloc]
initWithTarget:self action:@selector(onSwipe)];

swipe.numberOfTouchesRequired = 2;
swipe.direction =
UISwipeGestureRecognizerDirectionDown;
[self.view addGestureRecognizer:swipe];
```

Gesture Recognizers

- Tapping (any number of taps)
- Pinching in and out (for zooming a view)
- Panning or dragging
- Swiping (in any direction)
- **Rotating (fingers moving in opposite directions)**

UIRotationGestureRecognizer

```
UIRotationGestureRecognizer *rotate =
[[UIRotationGestureRecognizer alloc]
initWithTarget:self action:@selector(onRotate)];

[self.view addGestureRecognizer:rotate];
```

Gesture Recognizers

- Tapping (any number of taps)
- Pinching in and out (for zooming a view)
- Panning or dragging
- Swiping (in any direction)
- Rotating (fingers moving in opposite directions)
- **Long press (also known as “touch and hold”)**

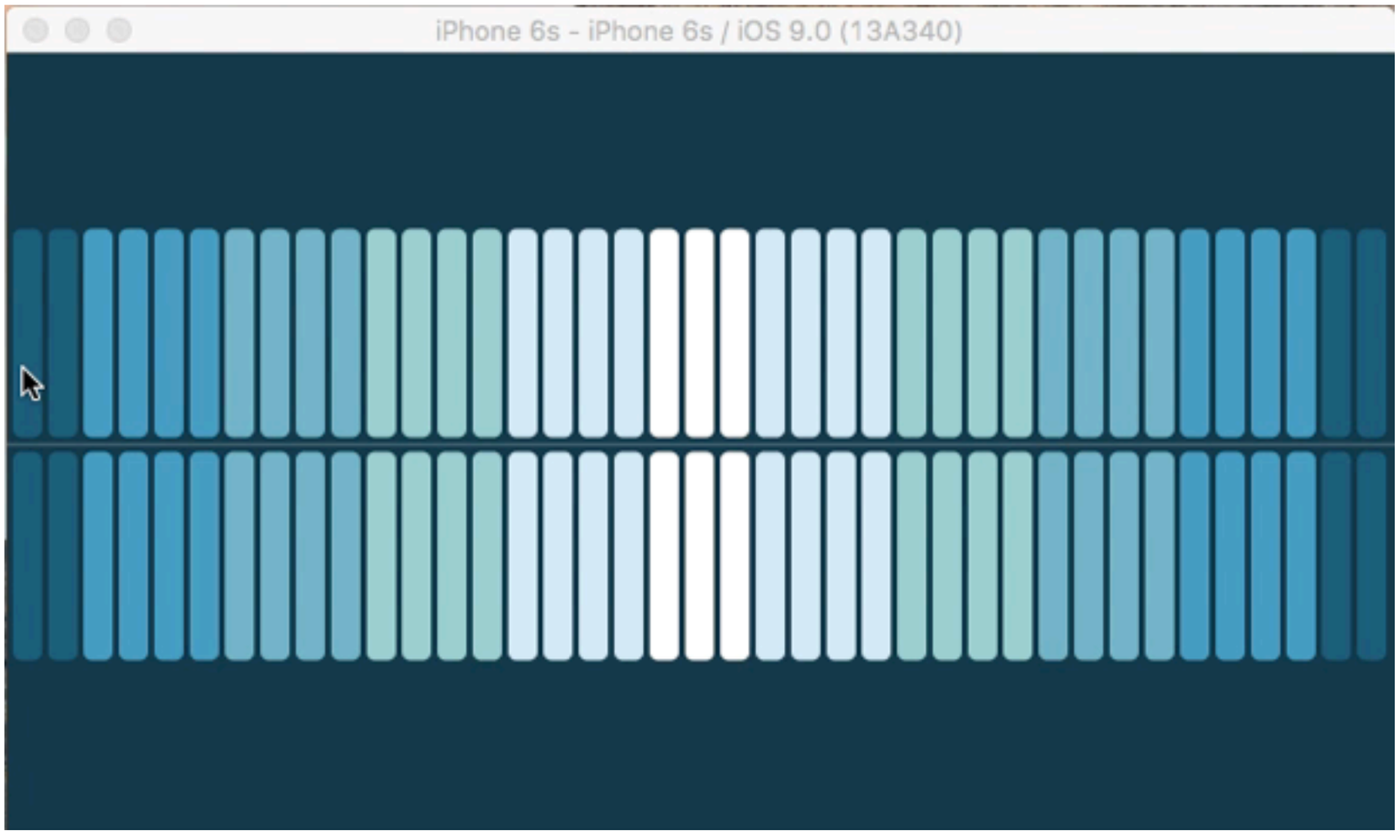
UILongPressGestureRecognizer

```
UILongPressGestureRecognizer *longPress =
[[UILongPressGestureRecognizer alloc]
initWithTarget:self
action:@selector(onLongPress)];

longPress.numberOfTapsRequired = 1;
longPress.numberOfTouchesRequired = 2;
longPress.minimumPressDuration = 1.f;
longPress.allowableMovement = 20;
[self.view addGestureRecognizer:longPress];
```

Homework

1. Realise design, presented in Equalizer.psd. All layers in psd-file should be UIView.
2. When we touch the location under some view line, we should change it's height, according to touch.
3. When we make double click on the screen, it should start some animation of coloured lines.

For example:



Useful links

- **Windows and Views:** https://developer.apple.com/library/prerelease/tvos/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/Introduction/Introduction.html#//apple_ref/doc/uid/TP40009503-CH1-SW2
- **About Events:** https://developer.apple.com/library/prerelease/tvos/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/Introduction/Introduction.html#//apple_ref/doc/uid/TP40009541-CH1-SW1