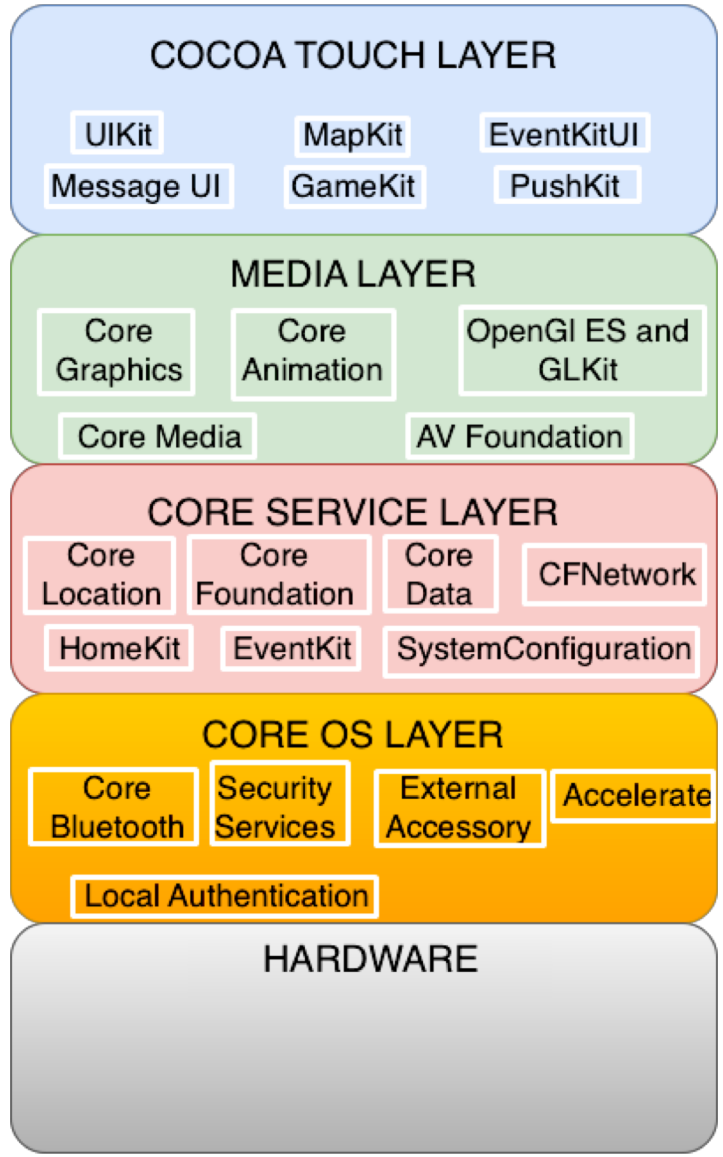
iOS

davide morelli

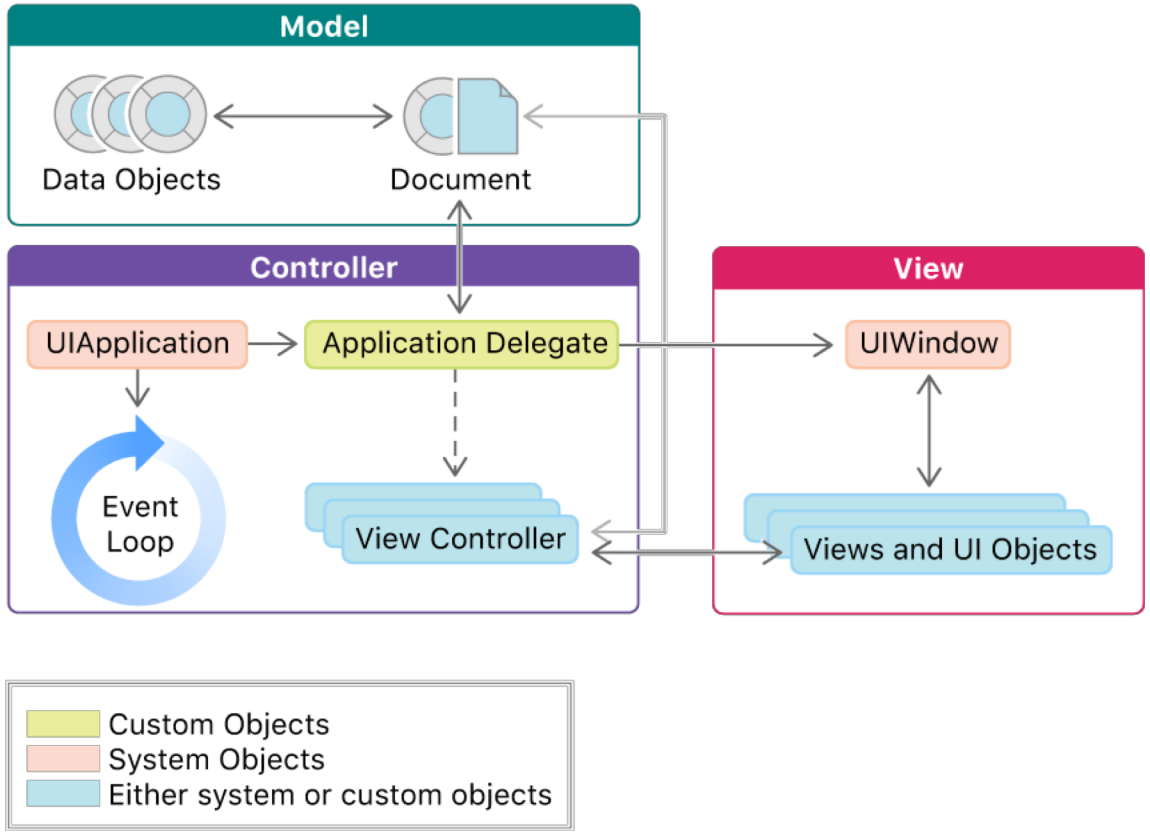
CI 2018


COCOA TOUCH LAYER
UIKit
MapKit
EventKitUI
Message UI
GameKit
PushKit
MEDIA LAYER
Core Graphics
Core Animation
OpenGl ES and GLKit
Core Media
AV Foundation
CORE SERVICE LAYER
Core Location
Core Foundation
Core Data
CFNetwork
HomeKit
EventKit
SystemConfiguration
CORE OS LAYER
Core Bluetooth
Security Services
External Accessory
Accelerate
Local Authentication
HARDWARE

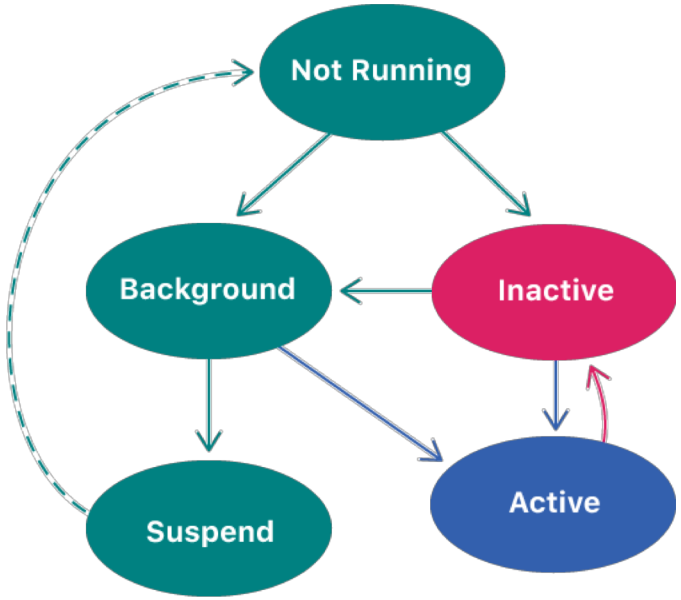
iOS frameworks

- https://developer.apple.com/documentation/
- UIKit https://developer.apple.com/documentation/uikit
- WatchKit https://developer.apple.com/documentation/watchkit
- Core Graphics https://developer.apple.com/documentation/coregraphics
- Core Image https://developer.apple.com/documentation/coreimage
- SpriteKit https://developer.apple.com/documentation/spritekit
- SceneKit https://developer.apple.com/documentation/scenekit
- GLKit/OpenGL ES
- Metal https://developer.apple.com/documentation/metal
- …

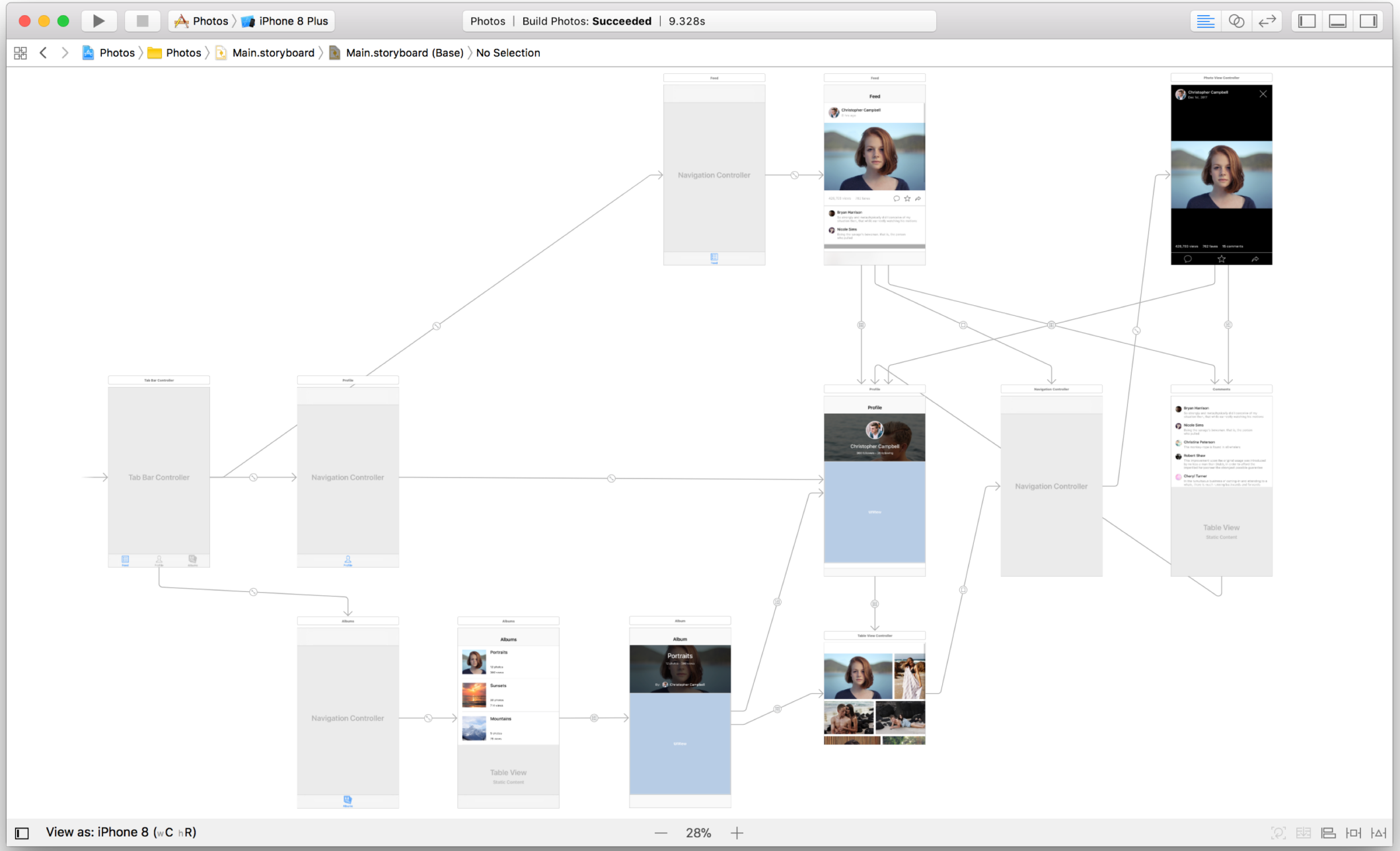
the core app objects



app lifecycle

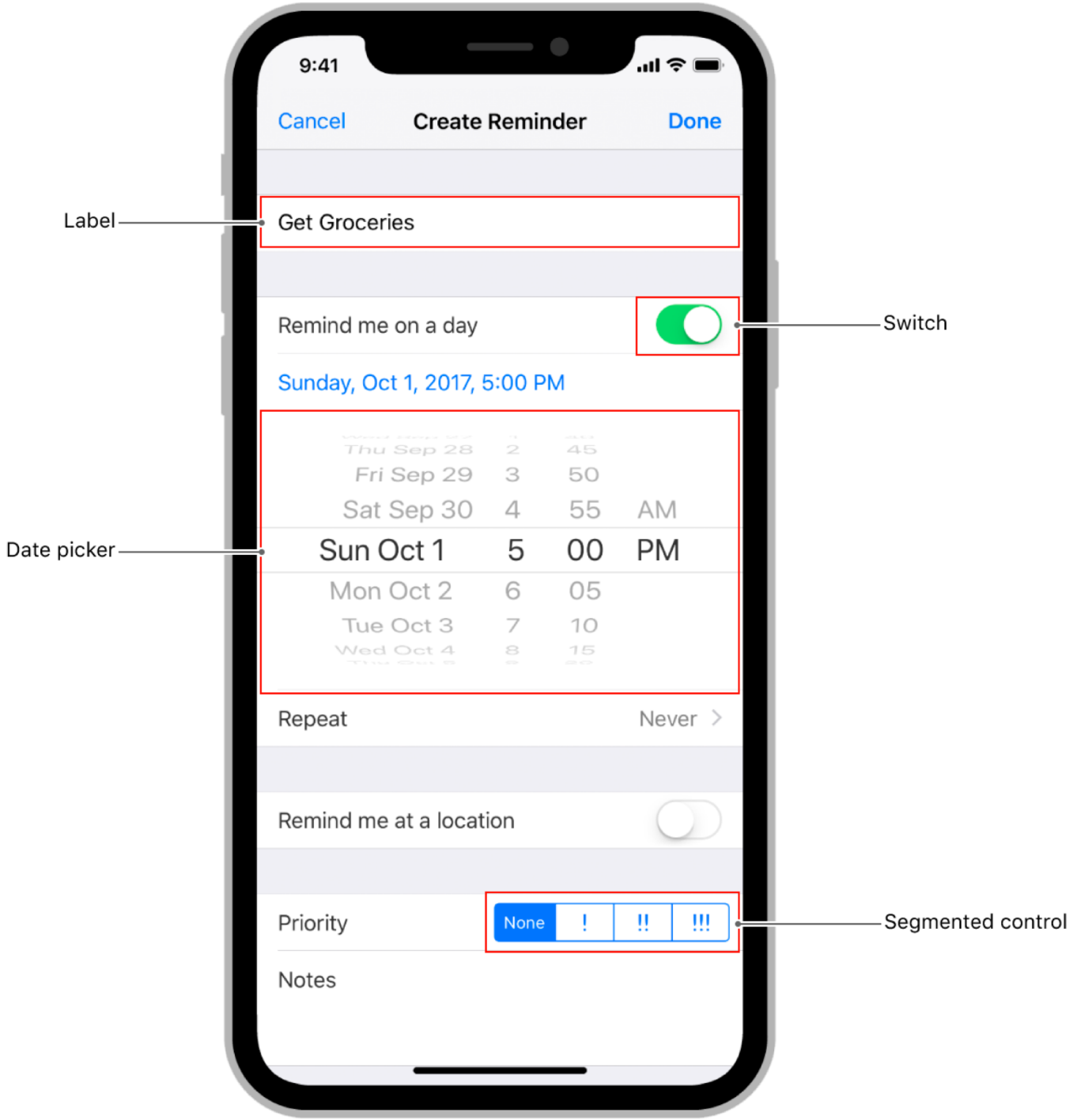


storyboard


Photos 〉 iPhone 8 Plus
Photos | Build Photos: Succeeded | 9.328s
Photos 〉 Photos 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
Navigation Controller
Feed
Christopher Campbell
Tab Bar Controller
Navigation Controller
Profile
Christopher Campbell
Navigation Controller
Table View
Static Content
Navigation Controller
Albums
Portraits
Sunsets
Mountains
Table View
Static Content
Album
Portraits
Table View Controller
View as: iPhone 8 (wC hR)
28%

storyboard

- defines the UX/UI flow
  - UIViewControllers
    - that contain System and custom UIViews
      - constraints
    - connected with *segue* objects
    - define references and events in code

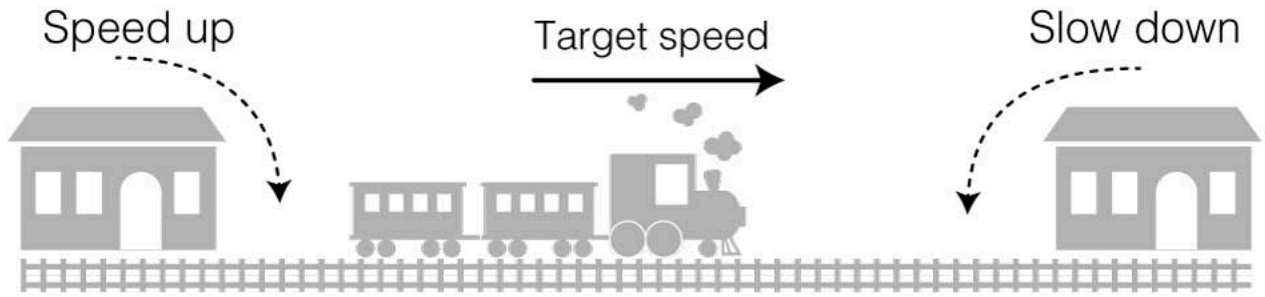


storyboard

- defines the UX/UI flow
  - UIViewControllers
    - that contain System and custom UIViews
      - constraints
    - connected with *segue* objects
    - define references and events in code
- example
  - UIViewController with a label and buttons
  - 2 UIViewControllers via segue from button
  - list of available system UIViews: UIKitCatalog
    - https://developer.apple.com/documentation/uikit/views_and_controls/uikitcatalog_creating_and_customizing_views_and_controls

Property based animations

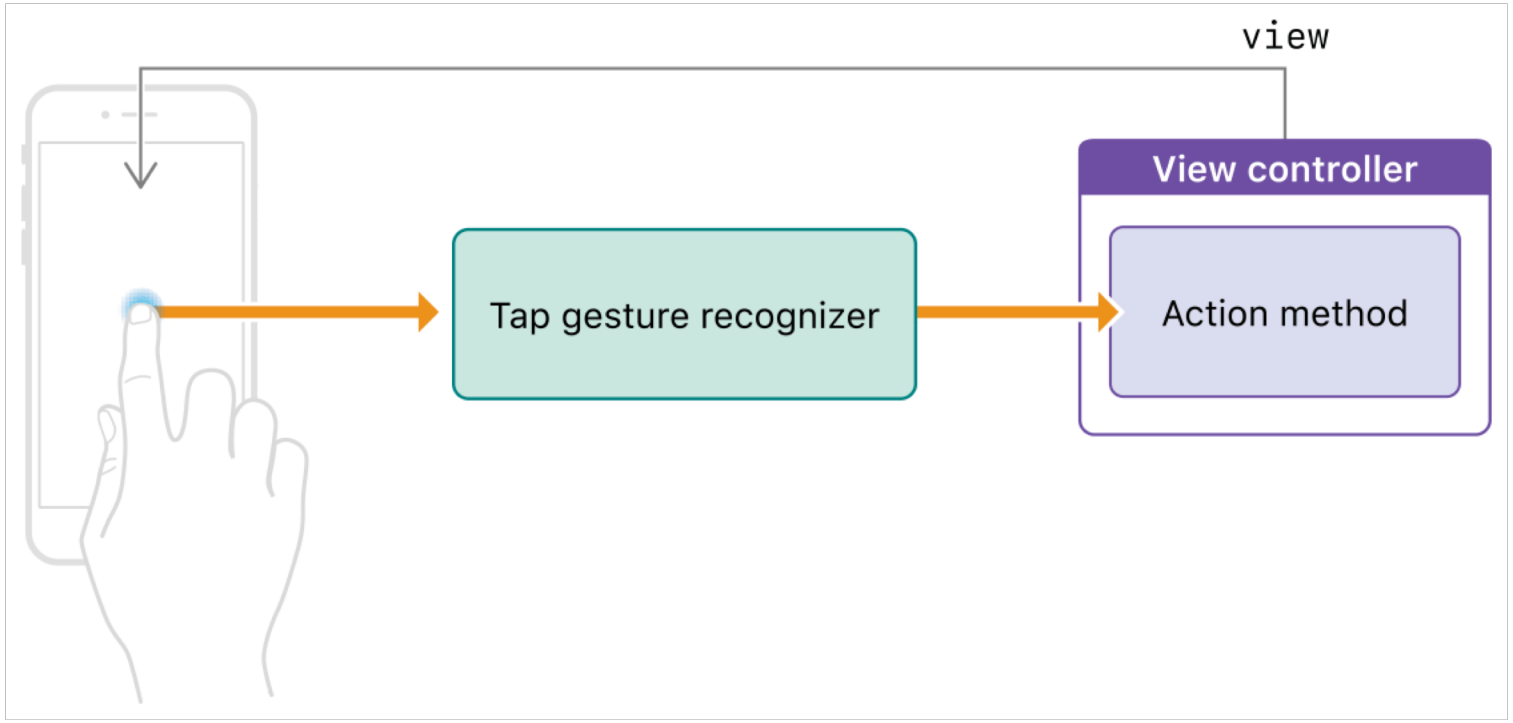
- UIView.animate(duration, [delay], [options], animations)
- options
  - repeat, autoreverse
- easing
- animations
  - set property to target value



- example https://www.raywenderlich.com/363-ios-animation-tutorial-getting-started

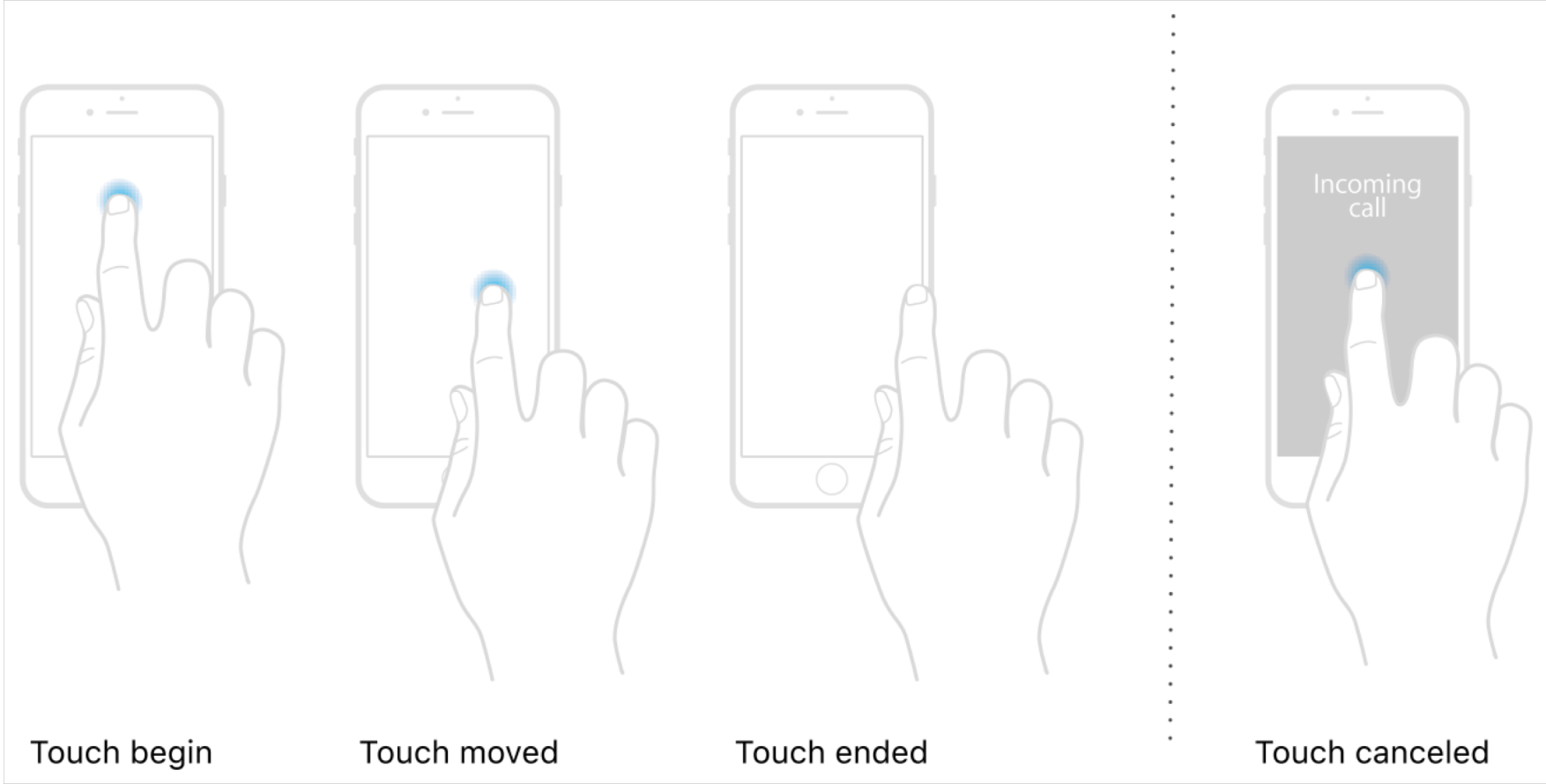
user interactions: touch, press, gestures

- UIKit gestures
  - tap
  - long press
  - pinch
  - pan
  - swipe
  - rotation
  - custom gestures



user interactions: touch, press, gestures

- touch events in UIView

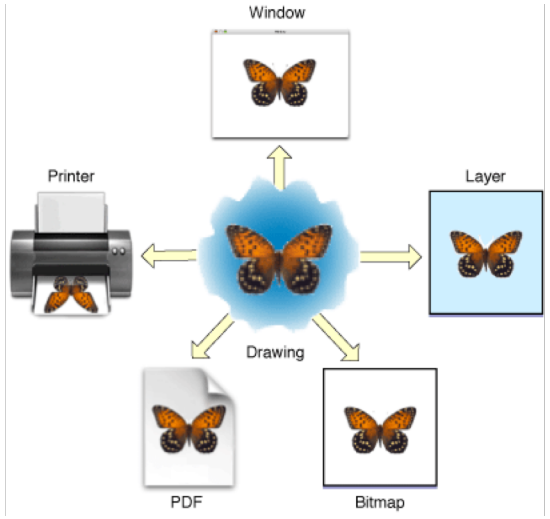


user interactions: other

- drag and drop
- pencil
- remote or controller
- force touch
- keyboard
- accessibility

core graphics

- graphic contexts
  - abstracts drawing operations from media differences
  - CGContextRef https://developer.apple.com/documentation/coregraphics/cgcontextref



opaque data types

- CGImageRef
- CGPathRef
- CGPatternRef
- CGLayerRef
- CGColorRef
- CGFontRef
- etc.

graphic context states

- stack of graphic states
- states include:
  - transformation matrix
  - clipping area
  - Line: width, join, cap, dash, miter limit
  - Color: fill and stroke settings
  - Anti-aliasing setting
  - Alpha value (transparency)
  - Text: font, font size, character spacing, text drawing mode
  - Blend mode

drawRect:

- override drawRect: of the UIView to draw, like the paint() in windows

```
- (void)drawRect:(CGRect)rect {
    // draw a square of random color
    CGContextRef ctx = UIGraphicsGetCurrentContext();
    CGContextSetRGBFillColor(ctx, arc4random()/RAND_MAX, arc4random()/RAND_MAX, arc4random()/RAND_MAX, 1.0);
    CGContextFillRect(ctx, CGRectMake(10.0, 10.0, 100.0, 100.0));
}
```

setNeedsDisplay()

- call setNeedsDisplay() on the instance of your custom view to invalidate and redraw

animating using timers

- see ball example