

# Let's make an App!

GeekPie Session

Tom Choi, Undergraduate, Senior.

You can distribute this keynote however you wish.

# Outline

macOS & iOS History

Objective-C & Swift Quick Look

Introducing Cocoa & Cocoa Touch

Build an iOS App with Xcode

# macOS & iOS History



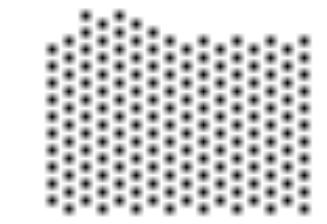
File Edit View Special

## Mac System Software

3 items

227K in disk

173K available



System Folder



Empty Folder



SysVersion



## System Folder

5 items

211K in folder

173K available



Finder



System



Imagewriter



Note Pad File



Scrapbook File

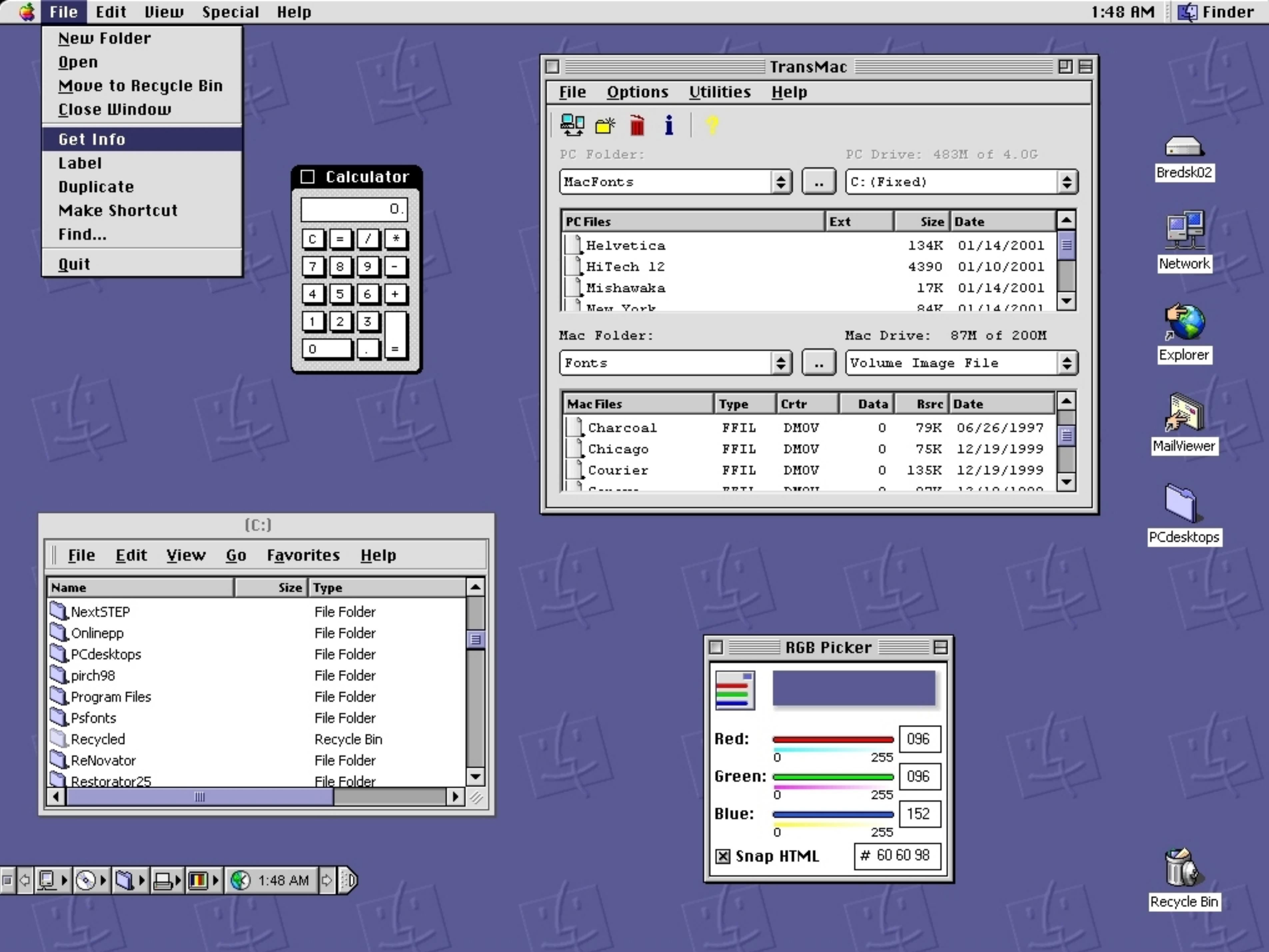


Clipboard File



Trash

System 1.0



Mac OS 9



NeXTSTEP





Macintosh HD

Install OS X



# OS X Mountain Lion

To set up the installation of OS X 10.8 Mountain Lion, click Continue.



Continue

OS X





Sierra

53,76 GB, 20,36 GB free

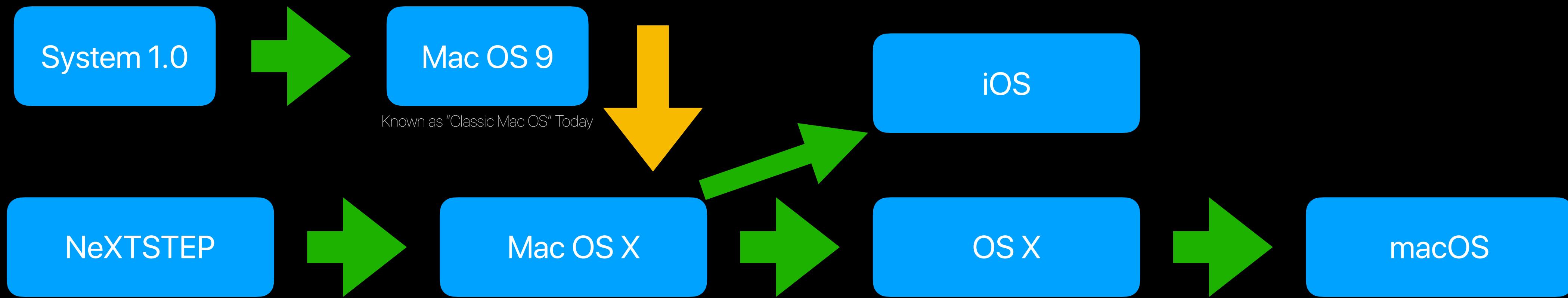
Downloads		Sierra	
		Search	
Name		Name	
Favorites		Favorites	
AirDrop		AirDrop	
All My Files		All My Files	
iCloud Dri...		iCloud Drive	
Applicatio...		Applications	
Desktop		Desktop	
Documents		Documents	
Downloads		Downloads	
Movies		Movies	
Music		Music	
Pictures		Pictures	
chris		chris	
Devices		Devices	
macOS Sierra		macOS Sierra	
Remote Disc		Remote Disc	
Shared		Shared	
Tags		Tags	
macOS Sierra Contacts.png		Applications	
macOS Sierra iTuns lyrics.png		Utilities	
macOS Sierra Disk Utility.png		App Store.app	
macOS Sierra Console.png		Apple Con...rator 2.app	
macOS Sierra Tabs in TextEdit.png		Automator.app	
macOS Sierra apple Music.png		Calculator.app	
macOS Sierra Siri file search.jpg		Calendar.app	
macOS Sierra widgets.jpg		Chess.app	
macOS Sierra notifications.jpg		Contacts.app	
macOS Sierra Messages URL inline.png		Dashboard.app	
macOS Sierra Messages Weather Channel.png		Dictionary.app	
macOS Sierra Messages YouTube inline.png		DVD Player.app	
macOS Sierra autocorrect settings.png		FaceTime.app	
macOS Sierra accessibility vision improvements.png		Font Book.app	
macOS Sierra Dwell Control.png		iBooks.app	
		Image Capture.app	
		iTunes.app	
		Launchpad.app	
		Mail.app	
		Maps.app	
		Messages.app	
		Mission Control.app	
		Notes.app	
		Photo Booth.app	
		Photos.app	
		Preview.app	
		QuickTime Player.app	
		Reminders.app	
		Safari.app	

macOS





# The Evolution of Apple OS



# Objective-C & Swift Quick Look

# Objective-C

C

Objective

```
.h
@interface GLKitViewController : GLKViewController
@end

.m
@interface GLKitViewController () {
    @property (strong, nonatomic) EAGLContext *context;
}
@end

@implementation GLKitViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    self.context = [[EAGLContext alloc]
initWithAPI:kEAGLRenderingAPIOpenGLGLES3];
    if (!self.context) {
        NSLog(@"Failed to create ES context");
    }
    GLKView *view = (GLKView *)self.view;
    view.drawableDepthFormat = GLKViewDrawableDepthFormat24;
    [EAGLContext setCurrentContext:self.context];
    NSString *path = [[NSBundle mainBundle] bundlePath];
}
@end
```

# Swift

Released in 2014

Fast, safe & modern

The latest version is Swift 4.1



```
- (void)viewDidLoad {
    [super viewDidLoad];
self.context = [[EAGLContext alloc] initWithAPI:kEAGLRenderingAPIOpenGL ES3];
if (!self.context) {
    NSLog(@"Failed to create ES context");
    return
}
GLKView *view = (GLKView *)self.view;
view.drawableDepthFormat = GLKViewDrawableDepthFormat24;
[EAGLContext setCurrentContext:self.context];
NSString *path = [[NSBundle mainBundle] bundlePath];
}
```

```
override func viewDidLoad() {
    super.viewDidLoad()
    guard let tmpContext = EAGLContext(api: .openGLES3) else{
        NSLog("Failed to create ES context")
        return
    }
    context = tmpContext
    let view = self.view as! GLKView
    view.drawableDepthFormat = .format24
    EAGLContext.setCurrent(context!)
    let _ = Bundle.main.bundlePath
}
```

# Swift Language Reference

[https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/Objective](https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/Objective)

<http://www.swift51.com/swift4.0/>

```
import Foundation
// 声明常量和变量，Swift 具有类型判断的能力
let anImmutableInt = 1 // = 两边必须有空格
var aMutableInt = 2
let someFloat:Float = 2 //可以指定声明类型
var aString = "233"

// 修改值
anImmutableInt = 8 // 错误，let 声明的变量不能修改其值
aMutableInt = 8
```

```
import Foundation
// 运算符号
let x:Double = 1
let y:Double = 2
let z = pow((x + y),3.0) / 2.0
print(String(z))
```

```
import Foundation
// While 循环, 循环内循环变量可变
var condA = true
while condA{
    condA = false
    print("Loop One time only")
}
repeat {
    condA = false
    print("Loop One time only")
}while condA
```

```
// While 循环, 循环内循环变量可变
for i in 0...5 {
    print(i) // Print 0 1 2 3 4 5
}

for i in 0..<5 {
    print(i) // Print 0 1 2 3 4
}
```

```
import Foundation
// 声明与使用函数，支持重载
func remainder(_ A:Double,division:Double)->Double{
    return A.truncatingRemainder(dividingBy: division)
    // return a % division
}

let c = remainder(10, division: 3.3)
print(c)
```

```
import Foundation
// 闭包的使用方法之一
func applyRulesToArray(input:[Double], _ rules:(Double)->(Double))->[Double]{
    var output = Array<Double>(repeating: 0.0, count: input.count)
    for i in 0..
```

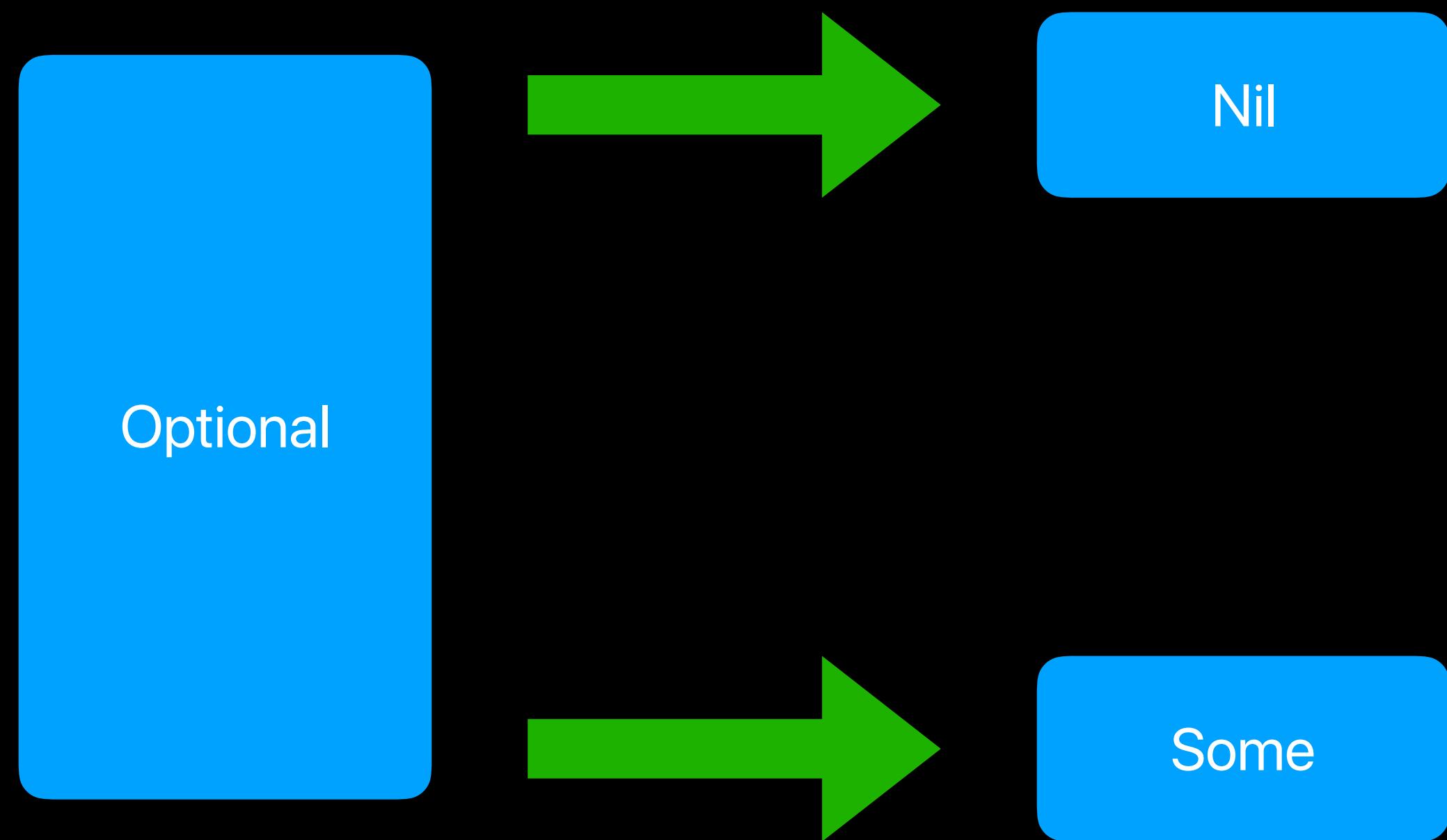
```
import Foundation
// 这里举一个例子就让Person继承NSObject类。
class Person: NSObject {
    // 声明存储属性, private 关键字使得属性无法在外部访问
    public var birth: Date
    // 声明计算属性, 必须使用var关键字
    var seconds: TimeInterval{
        return Date().timeIntervalSince(birth)
    }
    // 修改出生日期, 函数
    func changeBirth(_ date: Date){
        birth = date
    }
    // 静态方法, 可以直接使用类名调用
    static func secondsSinceBirth(_ date: Date) -> TimeInterval{
        return Date().timeIntervalSince(date)
    }
    // 类构造器
    init(_ date: Date) {
        birth = date
    }
}
```

```
let dateFormatter = DateFormatter()
dateFormatter.dateFormat = "yyyy-MM-dd"
let he: Person = Person(dateFormatter.date(from: "1995-03-15")!)
he.changeBirth(dateFormatter.date(from: "1926-08-17")!)
print("He lived for \(he.seconds) seconds.")

let secondsPassed =
Person.secondsSinceBirth(dateFormatter.date(from: "1970-01-01")!)
print("\(secondsPassed) seconds passed since 1970")
```

# Optional Type

Optional 类型可能包含一个值，也有可能没有值,类似于 Rust 的 Option



```
import Foundation
var a:Int? // Int Optional
a = nil // Optional 可以是空的
a = 1 // Optional 可以赋值
let b = a // b 也是 Int Optional (Int?)
let c = b! //! 对Optional进行拆包, c为Int, 如果b=nil那么会报错

guard let foo = b else {
    // b = nil 时会进入到这里
    exit(1)
}
print(foo) // foo 是Int类型而不是Optional

if let foo = b {
    print(foo) // foo 是Int类型而不是Optional
}else{
    // b = nil 时会进入到这里
    exit(1)
}
```

```
import Foundation
extension Int{
    var square:Int {
        return Int(pow(Double(self),2.0))
    }
}
print(32.square) // print 1024
```

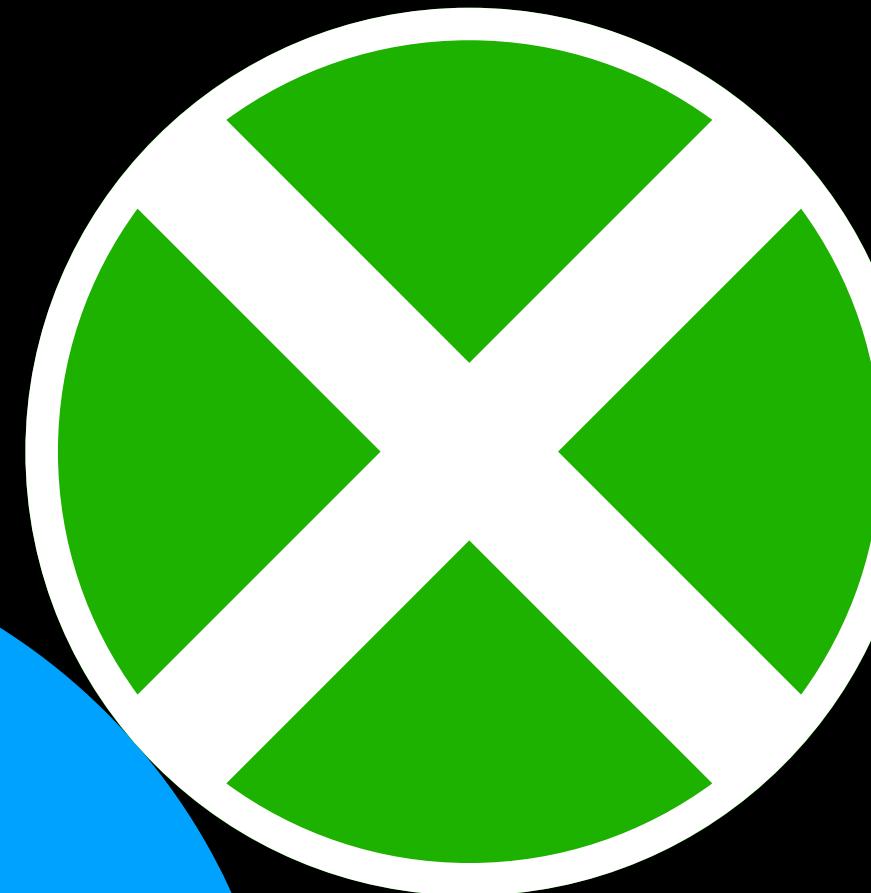
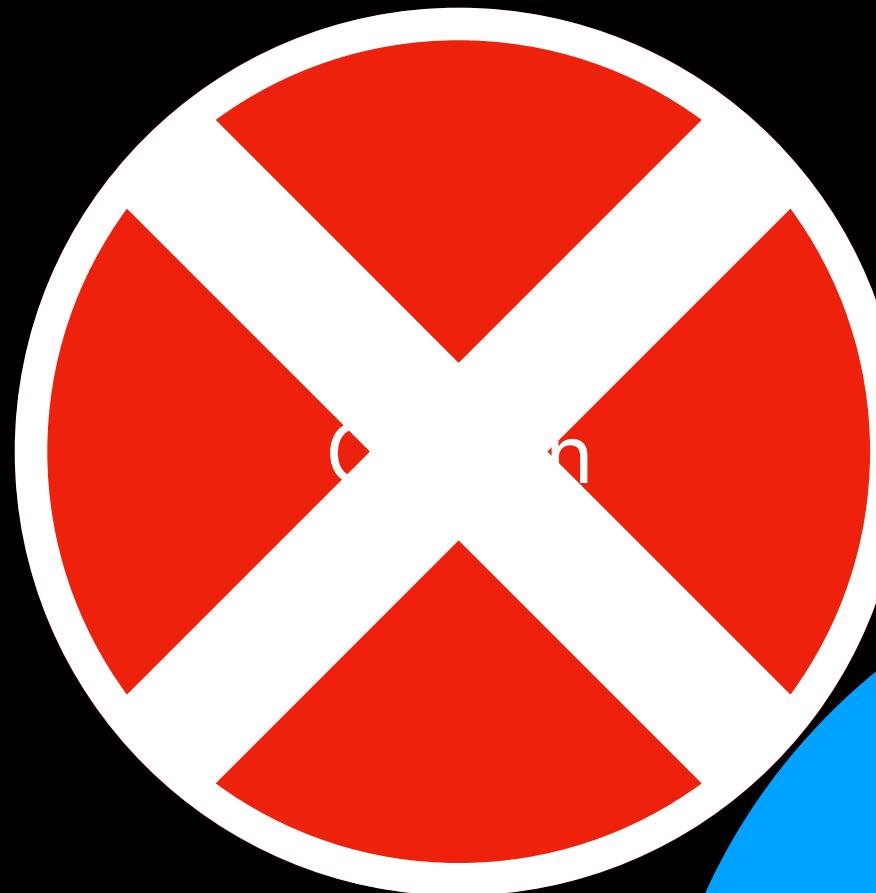
```
import Foundation
// inout 关键字将值类型以引用方式传递，这样可以修改函数外部参数的值
// 有点类似于 C# ref/out
// 范形的使用方式之一
func swapTwoValue<T>(a: inout T, b: inout T) {
    let tempValue = a
    a = b
    b = tempValue
}
var x = 1; var y = 2; swapTwoValue(a: &x, b: &y); print(x,y) // print 2
1
var s1 = "123"; var s2 = "321"; swapTwoValue(a: &s1, b:
&s2); print(s1,s2)
// print 321 123
```

# Introducing Cocoa & Cocoa Touch

# 可...可可粉?

A Native API for macOS and iOS

Cocoa



GymKit

SpriteKit

HomeKit

ARKit

UIKit

HomeKit

Grand Central Dispatch

Core Animation

AVFoundation

HealthKit

Metal

SceneKit

# Outline

- macOS & iOS History
- Objective-C & Swift Quick Look
- Introducing Cocoa & Cocoa Touch
- **Build an iOS App with Xcode**



# Welcome to Xcode

Version 9.2 (9C40b)

-  **Get started with a playground**  
Explore new ideas quickly and easily.
-  **Create a new Xcode project**  
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.
-  **Clone an existing project**  
Start working on something from an SCM repository.

Show this window when Xcode launches [Open another project...](#)

ARFACE  
TimeCapsule

HairRendering  
Personal

SwiftDemo  
~/Desktop

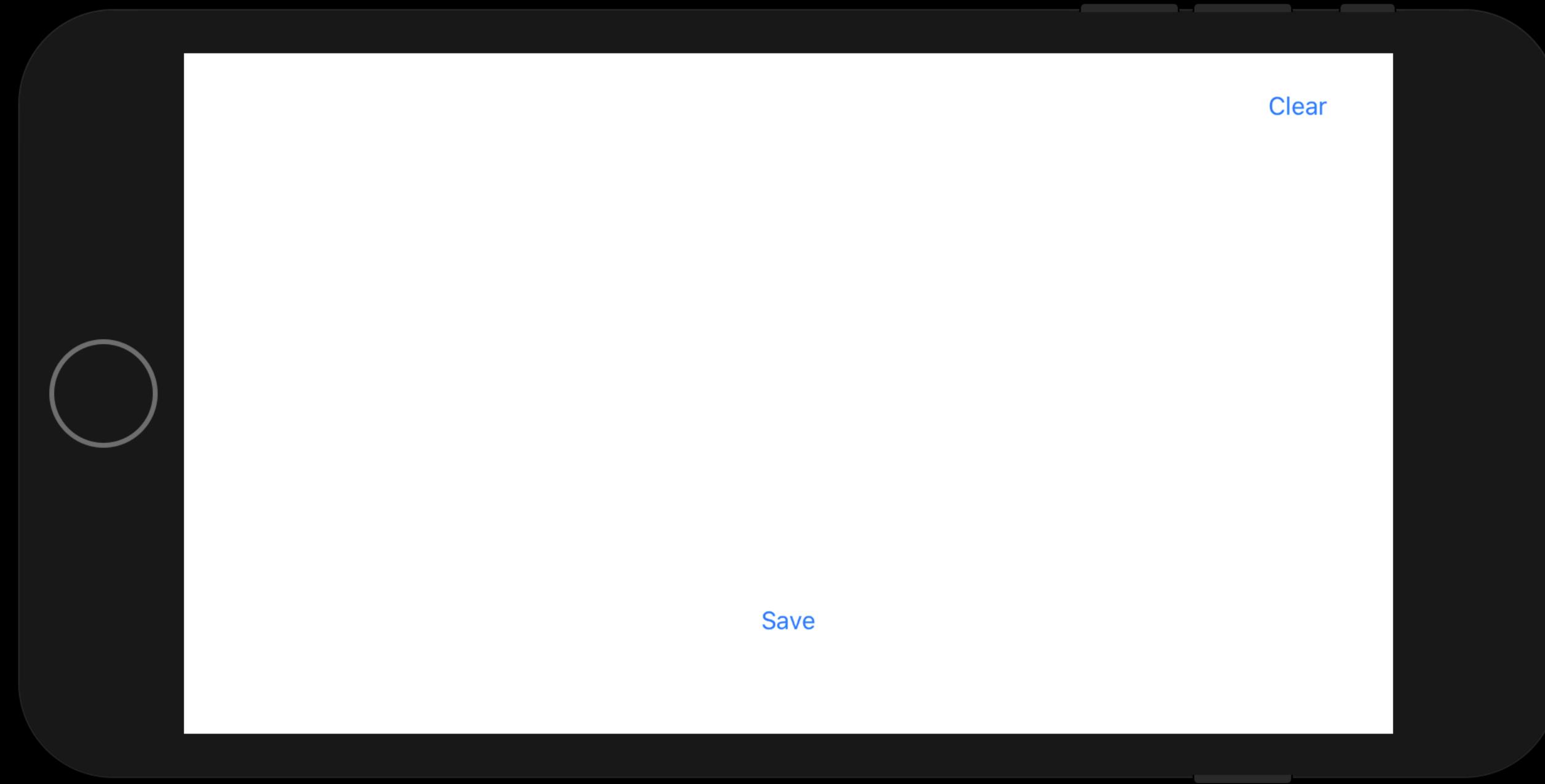
Demo  
~/Desktop

ARDavinci  
~/Developer

ARViewProj  
~/Developer

glut  
~/Developer

ray  
~/Developer



iPhone 8 Plus - 11.2

**<https://github.com/TomJinW/SketchPadExample>**

# Demo