Miscellaneous Topics

Mobile Application Development in iOS
School of EECS
Washington State University
Instructor: Larry Holder
Outline

• Access app info
• Rename app and Xcode project
• Add app icons
• Package management with CocoaPods
  – Example: Debugging with Crashlytics
• Detect if running in iOS simulator
• HomeKit
Access App Info
Access App Info.plist

Mobile Application Development in iOS
Access App Info.plist

// Get app name
let appName = Bundle.main.object(forInfoDictionaryKey: "CFBundleName") as! String

// Get app version
let appVersion = Bundle.main.object(forInfoDictionaryKey: "CFBundleShortVersionString") as! String

// Get app build
let appBuild = Bundle.main.object(forInfoDictionaryKey: "CFBundleVersion") as! String
Rename App and Xcode Project
Rename App Display Name

- Add "Bundle display name" key to Info.plist
Rename Xcode Project

• Step 1: Change project name in upper left of file hierarchy
  – Rename project content items
Rename Xcode Project

• Step 2: Rename scheme
Renaming Xcode Project

- Step 3: Change bundle identifier
Renaming Xcode Project

• Step 4: Change top-level folder/file names

Change app name in code comments
Renaming Xcode Project

• Step 5: Change Info.plist file in build settings

Done!
Add App Icons
Add App Icons
Add App Icons
Add App Icons

• Create main high-resolution icon

• Use app icon service to generate different resolutions
  – E.g., www.makeappicon.com

• Drag and drop into AppIcon assets
Add App Icons

### File Structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Date Modified</th>
<th>Size</th>
<th>Kind</th>
</tr>
</thead>
<tbody>
<tr>
<td>android</td>
<td>Today at 12:48 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>imessenger</td>
<td>Today at 12:48 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>ios</td>
<td>Today at 12:48 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>AppIcon.appiconset</td>
<td>Today at 12:48 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>Contents.json</td>
<td>Nov 23, 2018 at 9:22 AM</td>
<td>3 KB</td>
<td>JSON File</td>
</tr>
<tr>
<td><a href="mailto:icon-App-20x20@1x.png">icon-App-20x20@1x.png</a></td>
<td>Today at 8:44 PM</td>
<td>1 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-20x20@2x.png">icon-App-20x20@2x.png</a></td>
<td>Today at 8:44 PM</td>
<td>3 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-20x20@3x.png">icon-App-20x20@3x.png</a></td>
<td>Today at 8:44 PM</td>
<td>5 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-29x29@1x.png">icon-App-29x29@1x.png</a></td>
<td>Today at 8:44 PM</td>
<td>2 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-29x29@2x.png">icon-App-29x29@2x.png</a></td>
<td>Today at 8:44 PM</td>
<td>4 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-29x29@3x.png">icon-App-29x29@3x.png</a></td>
<td>Today at 8:44 PM</td>
<td>8 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-40x40@1x.png">icon-App-40x40@1x.png</a></td>
<td>Today at 8:44 PM</td>
<td>3 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-40x40@2x.png">icon-App-40x40@2x.png</a></td>
<td>Today at 8:44 PM</td>
<td>7 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-40x40@3x.png">icon-App-40x40@3x.png</a></td>
<td>Today at 8:44 PM</td>
<td>13 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-60x60@2x.png">icon-App-60x60@2x.png</a></td>
<td>Today at 8:44 PM</td>
<td>13 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-60x60@3x.png">icon-App-60x60@3x.png</a></td>
<td>Today at 8:44 PM</td>
<td>24 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-76x76@1x.png">icon-App-76x76@1x.png</a></td>
<td>Today at 8:44 PM</td>
<td>6 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-76x76@2x.png">icon-App-76x76@2x.png</a></td>
<td>Today at 8:44 PM</td>
<td>18 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-83.5x83.5@2x.png">icon-App-83.5x83.5@2x.png</a></td>
<td>Today at 8:44 PM</td>
<td>21 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:iTunesArtwork@1x.png">iTunesArtwork@1x.png</a></td>
<td>Today at 8:44 PM</td>
<td>384 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:iTunesArtwork@2x.png">iTunesArtwork@2x.png</a></td>
<td>Today at 8:44 PM</td>
<td>133 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:iTunesArtwork@3x.png">iTunesArtwork@3x.png</a></td>
<td>Today at 8:44 PM</td>
<td>709 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td>README.md</td>
<td>Apr 29, 2016 at 1:18 PM</td>
<td>1 KB</td>
<td>Markdown Document</td>
</tr>
</tbody>
</table>

**Mobile Application Development in iOS**
Add App Icons

Mobile Application Development in iOS

18
Package Management with CocoaPods
Package Management with CocoaPods

- [https://cocoapods.org](https://cocoapods.org)
- MacOS installation (using Ruby)
  - `sudo gem install cocoapods`
- Create Podfile in top-level app directory
  - `pod init`
- Add pods to Podfile
- Install dependencies
  - `pod install`
CocoaPods Example: Crashlytics

• How to know when your deployed app crashes?


• Free!
Podfile Example: Crashlytics

# Podfile

target 'TestApp2' do
  use_frameworks!

  pod 'Fabric'
  pod 'Crashlytics'

end

> pod install
...
[!] Please close any current Xcode sessions and use 'TestApp2.xcworkspace' for this project from now on.
CocoaPods

- Usually requires an import and initialization
Crashlytics Extra Steps (1)

- Add a Run Script Build Phase
  - Add Info.plist path to Input Files
Crashlytics Extra Steps (2)

- Add API key to info.plist
Crashlytics Example

• Crash the app

```swift
import Crashlytics

@IBAction func crashAppTapped(_ sender: UIButton) {
    // Store some user info
    Crashlytics.sharedInstance().setUserIdentifier("12345")

    // Crash the app
    Crashlytics.sharedInstance().crash()
}
```
Crashlytics Example

Email

NEW ISSUE

Heads up! We detected a new fatal issue in ViewController.swift line 22

See what went wrong

Dashboard

#1
ViewController.swift line 22
ViewController.crashAppTapped(_:)

Device
iPhone 6
iOS 12.1.4 (16D57)
Portrait
No

Operating System
iOS
Portrait
No

Device Statistics
2%

Keys
No keys found. Learn more

Stacktrace
Crashed: com.apple.main-thread
EXC_BREAKPOINT 0x0000000010021a68

0
TestApp2
<Crashlytics crash> + 14

1
TestApp2
ViewController.swift line 22
ViewController.crashAppTapped(_:)

2
TestApp2
<compiler-generated> line 0
@test ViewController.crashAppTapped(_:)

3
UIKitCore
UIApplication sendActionToMainThread() + 96

5
UIKitCore
UIApplicationMain + 212

20
TestApp2
AppDelegate.swift line 14
main

21
libdyld.dylib
start + 4
Other Crash Tools

- **Firebase (Android, iOS)**
  - [firebase.google.com](https://firebase.google.com)

- **Xcode (iOS)**
  - Once app on App Store or Test Flight
Other Package Management Tools

• Carthage
  – https://github.com/Carthage/Carthage
  – Similar to CocoaPods, but decentralized

• Swift Package Manager
  – https://swift.org/package-manager/
  – Doesn’t work with iOS (yet)
Detecting if Running in iOS Simulator
Create Swift Flag in Build Settings
extension UIDevice {
    var isSimulator: Bool {
        #if IOS_SIMULATOR // flag defined in build settings
            return true
        #else
            return false
        #endif
    }
}

// Usage
if (UIDevice.current.isSimulator) {
    // Do this if app running in simulator
} else {
    // Do this if app running on device
}
HomeKit
HomeKit

• Framework for monitoring and control of connected accessories in user’s home
HomeKit

• Add HomeKit to app’s Capabilities
  – Requires developer account

• Requires description for why app needs access to HomeKit
Home Layout

• Use `HMHome` class to create home layout
• Each home consists of rooms (`HMRoom`)
• Each room consists of accessories (`HMAccessory`)
  – E.g., lamp
• Each accessory provides one or more services (`HMService`)
  – E.g., turn lamp on or off
• Each service can have one or more characteristics (`HMCharacteristic`)
  – E.g., on/off status of lamp
• Rooms can be grouped into zones (`HMZone`)
Accessory Types

• Predefined services and characteristics for common accessories
  – Light and power
  – Air, temperature, humidity and water
  – Locks, safety and security
  – Cameras and sound

• Other accessory types possible
  – Defined by the manufacturer of the accessory
  – Accessory must include HomeKit-compatible hardware
    • Complies with “Made For iOS” (MFi) program
Events and Triggers

• Events
  – Location events
  – Time events
  – Characteristic events
  – Presence events

• Trigger
  – Predicate that determines if trigger should fire
  – Scene (array of Events) executed when predicate is true

E.g., “When the front door is opened, turn on the hallway light.”
Home Manager

- HMHomeManager
Home Manager Delegate

- HMHomeManagerDelegate
  - homeManagerDidUpdateHomes
  - homeManagerDidUpdatePrimaryHome
  - didAdd (home)
  - didRemove (home)
import HomeKit

class ViewController: UIViewController, HMHomeManagerDelegate {
    var homeManager = HMHomeManager()

    override func viewDidLoad() {
        super.viewDidLoad()
        homeManager.delegate = self
    }

    // Called after initial fetch of home data, or any changes
    func homeManagerDidUpdateHomes(_ manager: HMHomeManager) {
        if (homeManager.homes.isEmpty) {
            print("no homes")
        } else {
            for home in homeManager.homes {
                print("Found home: \(home.name)")
            }
        }
    }
}
Testing HomeKit App

• HomeKit Accessory Simulator
  – Don’t need an actual HomeKit accessory to test
  – Download by clicking on “Download HomeKit Simulator” in Capabilities pane
    • Find latest version of “Hardware IO Tools for Xcode”
    • Install simulator in Applications folder
Resources

• developer.apple.com/homekit/

• HomeKit Developer Guide
  – developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/HomeKitDeveloperGuide