Miscellaneous Topics

Mobile Application Development in iOS

School of EECS

Washington State University

Instructor: Larry Holder
• Access app info
• Rename app and Xcode project
• Add app icons *(updated 1/17/20)*
• Package management with CocoaPods *(updated 3/13/20)*
  – 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
Add App Icons
Add App Icons
Add App Icons

Mobile Application Development in iOS
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Date Modified</th>
<th>Size</th>
<th>Kind</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsu-logo-700x700</td>
<td>Today at 12:43 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>android</td>
<td>Today at 12:42 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>imessenger</td>
<td>Today at 12:42 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>ios</td>
<td>Today at 12:45 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>AppIcon.appIconset</td>
<td>Today at 12:42 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>Contents.json</td>
<td>Nov 23, 2018 at 8:22 AM</td>
<td>3 KB</td>
<td>JSON Document</td>
</tr>
<tr>
<td><a href="mailto:icon-App-20x20@1x.png">icon-App-20x20@1x.png</a></td>
<td>Today at 8:39 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 8:39 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 8:39 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 8:39 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 8:39 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 8:39 PM</td>
<td>7 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 8:39 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 8:39 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 8:39 PM</td>
<td>11 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-App-60x60@1x.png">icon-App-60x60@1x.png</a></td>
<td>Today at 8:39 PM</td>
<td>11 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 8:39 PM</td>
<td>19 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 8:39 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 8:39 PM</td>
<td>14 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 8:39 PM</td>
<td>17 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:ItunesArtwork@2x.png">ItunesArtwork@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>238 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:ItunesArtwork@1x.png">ItunesArtwork@1x.png</a></td>
<td>Today at 8:39 PM</td>
<td>100 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:ItunesArtwork@2x.png">ItunesArtwork@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>238 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:ItunesArtwork@3x.png">ItunesArtwork@3x.png</a></td>
<td>Today at 8:39 PM</td>
<td>411 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td>README.md</td>
<td>Apr 29, 2016 at 12:18 PM</td>
<td>1 KB</td>
<td>Markdown Document</td>
</tr>
<tr>
<td>watchkit</td>
<td>Today at 12:47 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>AppIcon.appIconset</td>
<td>Today at 12:42 PM</td>
<td>--</td>
<td>Folder</td>
</tr>
<tr>
<td>Contents.json</td>
<td>Sep 15, 2016 at 6:58 AM</td>
<td>2 KB</td>
<td>JSON Document</td>
</tr>
<tr>
<td><a href="mailto:icon-24@2x.png">icon-24@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>3 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-27.5@2x.png">icon-27.5@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>4 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-29@2x.png">icon-29@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>4 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-29@3x.png">icon-29@3x.png</a></td>
<td>Today at 8:39 PM</td>
<td>7 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-40@2x.png">icon-40@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>7 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-44@2x.png">icon-44@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>7 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-66@2x.png">icon-66@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>18 KB</td>
<td>PNG image</td>
</tr>
<tr>
<td><a href="mailto:icon-88@2x.png">icon-88@2x.png</a></td>
<td>Today at 8:39 PM</td>
<td>23 KB</td>
<td>PNG image</td>
</tr>
</tbody>
</table>
Add App Icons
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`
- Open `.xcworkspace` version of project
CocoaPods Example: Crashlytics

• How to know when your deployed app crashes?

• [https://firebase.google.com/docs/docs/crashlytics](https://firebase.google.com/docs/docs/crashlytics)

• Free for small deployments
Firebase Console

- console.firebase.google.com
- Add project
- Add Firebase to your iOS app
Podfile Example: Crashlytics

> pod init

```bash
# Podfile

target 'CrashApp' do
  use_frameworks!

  pod 'Fabric'
  pod 'Crashlytics'
  pod 'Firebase/Analytics'

end
```

> pod install

...!

[!] Please close any current Xcode sessions and use ‘CrashApp.xcworkspace’ for this project from now on.
CocoaPods

• Import Firebase and initialization in AppDelegate
Crashlytics Extra Steps (1)

- Add a Run Script Build Phase
Crashlytics Extra Steps (2)

- Collect symbols
Crashlytics Example

• Add code to crash the app
  
  – Can collect various information

```swift
import Crashlytics

@IBAction func crashMeTapped(_ sender: UIButton) {

    // Store some user info
    Crashlytics.sharedInstance().setUserIdentifier("Larry")

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

• Crash the app
  – Run app from Xcode to install on simulator/device
  – Run app without Xcode directly on simulator/device
  – Crash the app
    • Crashlytics.sharedInstance().crash()
  – Run app again from Xcode (sends crash report)
  – View on Firebase Crashlytics dashboard
Crashlytics Example
Other Crash Tools

• **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/
Detecting if Running in iOS Simulator
Create Swift Flag in Build Settings
Extend UIDevice Class

```swift
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 is 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**)

Mobile Application Development in iOS
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


- HomeKit Developer Guide