CS 4220 Intro to iOS Programming

The World of iOS

macOS History

1983: Objective-C is released

- strict superset of C
- heavily influenced by Smalltalk, one of the earliest object-oriented languages

1985: Steve Jobs leaves Apple to found NeXT

1989: NeXT releases *NeXTSTEP*

- Graphical object-oriented OS
- Based on the Unix kernel
- Written in C and Objective-C
- Object-oriented APIs (Kits) for developers made with Objective-C

macOS History

1997: Apple acquires NeXT

- Primarily to use *NeXTSTEP* as the new operating system for Macs
- Before long, Steve Jobs becomes interim CEO

1999: Mac OS X Server 1.0

2000: Jobs becomes permanent CEO

2001: Mac OS X (10.0) for Consumers

Key macOS Frameworks/Tools

Project Builder was renamed *Xcode*

Developer Kit was renamed *Cocoa:*

- **App Kit**: classes needed to build macOS applications and GUIs
- **Foundation**: file persistence, text processing, date/time, networking, and types such as NSString, NSNumber, NSArray (NS stands for NeXTSTEP)
- **Core Data**: object graph and data persistence
- **Objective-C Runtime**: support for the dynamic properties of the Objective-C language
- **C Standard Library**

iOS History

- **2007**: First iPhone, iPod Touch, Apple TV with a limited version of OS X
- **Spring 2008**: First SDK for “iPhone OS 1”
- **Summer 2008**: iPhone OS 2 and App Store
- **2010**: Renamed to iOS; first iPad released; iOS 4
- **2014**: Apple’s new language, Swift, announced at WWDC
- **2015**: Apple Watch; Swift 2; iOS 9
- **Late 2015**: Swift was open-sourced
- **2017**: Swift 4 released
- **2019**: Swift 5.0 released, Apple announces SwiftUI at WWDC

Key iOS Frameworks/Tools

- **Cocoa Touch**: contains key frameworks for building iOS (UIKit, MapKit, GameKit, etc)
- **UIKit**: crucial infrastructure for implementing graphical, touch-based, event-driven apps in iOS
- **Foundation**, **CoreData**, **Obj-C Runtime**, and **C Standard Library** just like macOS
- **Device Simulator**
- **Xcode** (also used for macOS development)
- **Swift** and the **Swift Standard Library** (also for MacOS development)

Developing for macOS vs. iOS

- macOS runs on machines with **more powerful processors** and **more memory**
- macOS runs on machines with **larger** displays supporting **multiple windows**, a **mouse**, and a **keyboard**
- iOS runs on devices with a **touch screen**, **GPS**, **camera**, and **accelerometer**

Developing for macOS vs. iOS

- macOS apps are **developed** and **tested on a Mac**
- iOS apps are **developed on a Mac** and tested in a **Simulator** and **on provisioned devices**
- macOS apps can be distributed **by any means** (including the official Mac App Store)
- iOS apps must be distributed in the **official iOS App Store** or in a **licensed Enterprise App Store**

Developing for macOS vs. iOS

Interface Development and Use

- iOS supports multi-tasking but not multiple programs displaying multiple windows at same time
- Caveat: Some iPads running iOS 9+ have a Split Screen mode

Low Level OS Access

- macOS applications can access most low level utilities
- iOS applications can also access low level utilities but must also meet stricter requirements on resources (threads, memory, storage, processing)

Devices

- iOS runs on the iPhone, iPod Touch, and iPad
- iOS is also the basis for watchOS and tvOS
- Many different models, sizes, resolutions, and specs between the devices
- However, **all run the same iOS**
- When developing across device sizes/types, coding differences are typically only in the View layer and can be minimized

Development, Testing, and Deployment

- Developing for iOS **requires** a Mac with Xcode installed
- A new version of Xcode is typically released **every year** along with a new iOS version
- The Simulator comes with Xcode and simulates all the currently supported devices
- Apple retires support for older iOS version on a fairly consistent schedule, but the current Xcode will allow you to target all supported versions

Apple ID

- If you have an Apple ID, you can access a great deal of content for free:
    - Access to the Developer Library with tons of great documentation and videos
    - Download current and previous versions of Xcode and other tools
    - Load applications onto actual devices for testing

Apple Developer Program

Individual Developer Program ($99/year)

- Can sell apps on the App Store (after review process) and receive 70% of sales revenue
- Beta testing with TestFlight (up to 2000 external testers)
- Download beta iOS/macOS/Xcode releases

Enterprise Program ($299/year)

- Everything in Individual Developer Program
- Developer accounts for team members
- Distribute apps in a private App Store

Obtaining Xcode

- You must have an Apple ID (free) in order to download and install Xcode
- Check the Canvas site for instructions
- Apple Developer Portal: developer.apple.com