

# Supplementary

## Comparing Android to iOS

CSCI3310 Mobile Computing &  
Application Development

# Comparison between Android & iOS

- Programming Language
- Developer friendliness
- Application framework
- IDE User interface
- Performance
- Others

# Comparison between Android & iOS

- Language
  - iOS: Objective C or Swift
  - Android: Java or Kotlin
- Developer Friendly
  - Garbage collection is much more friendly to programmer than ownership handling of memory
  - Android is fully open sourced while iOS is closed source
  - iOS simplifies coding on certain instances, but at the cost of complicating /hindering usage in some other ways

# Comparison between Android & iOS

- User interface
  - Android is quickly catching up with huge improvement in UI
- Application framework
  - Android model an app as a set of Activities, Services & Content Providers
  - iOS sandboxes an app as a traditional application on an OS (MVC)
- Performance
  - To combat the efficiency issues with Java Virtual machine, ART is developed, NDK can also be used
  - iOS already taken advantage of C's high performance

# Android vs. iOS [App Development]

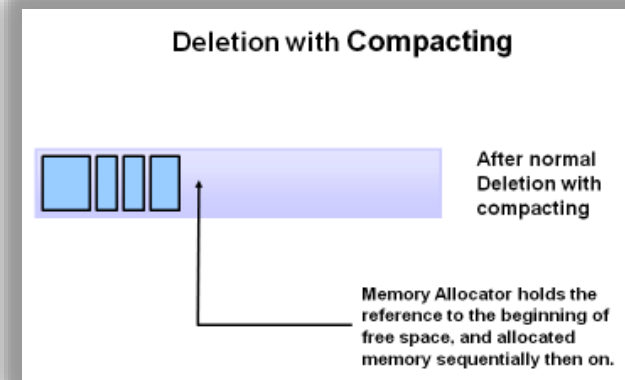
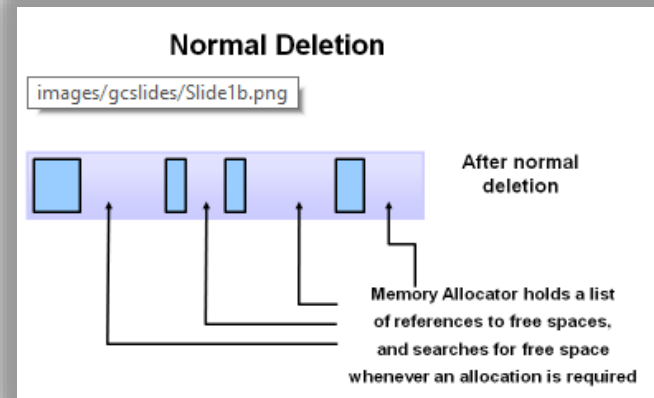
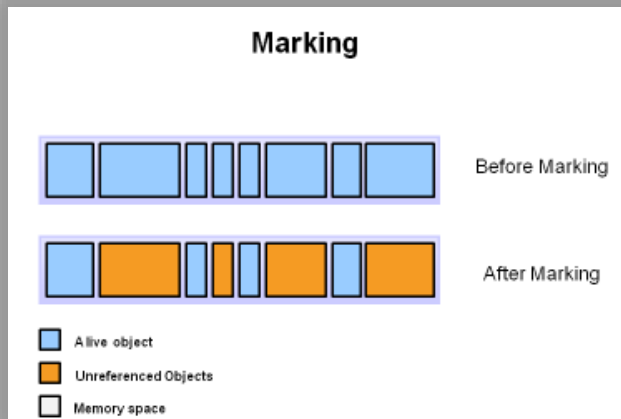
- Language

- **Objective C** has a more steep learning curve, especially when deal with memory management
- **Swift** is competitive as it got many new designs
- **Java** is more friendly as most programmers learnt Java before
- **Kotlin** is comparable to Swift in terms of the design

# Android vs. iOS [Development Friendliness]

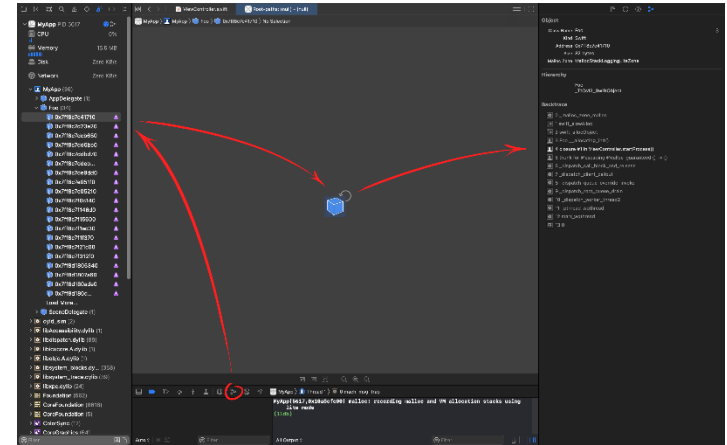
- Developer Friendliness

- Garbage collection is much more friendly to programmer than ownership handling of memory



# Android vs. iOS [Development Friendliness]

- Developer Friendliness
  - Memory management is the developer's job
  - Developers has to go through a painful process to turn a phone into development version in iOS
  - iOS simplifies coding on certain instances e.g. UIViews, but at the cost of complicating /hindering usage in the MVC architecture



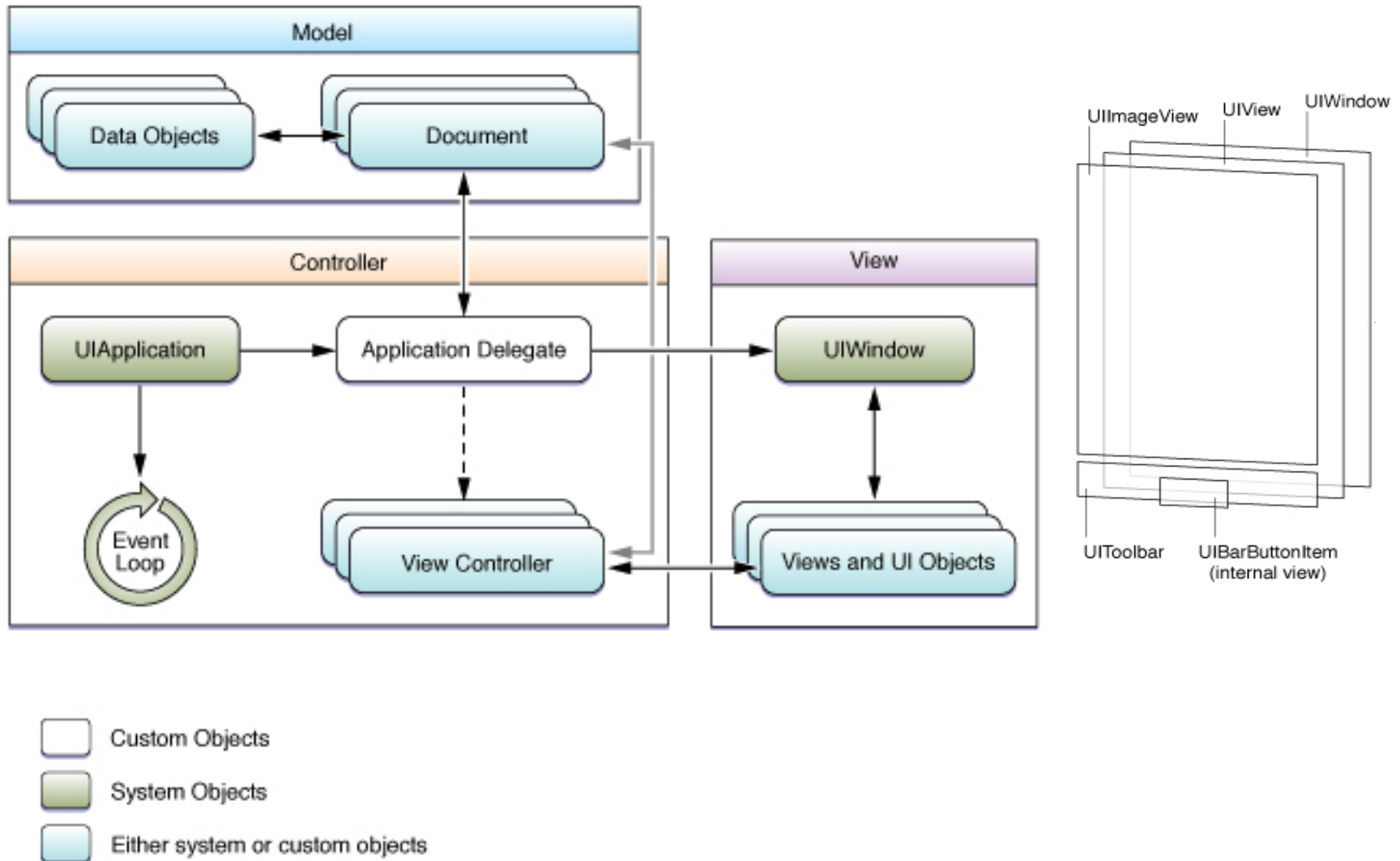
Debug Memory Graph  
in iOS app

# Android vs. iOS [App Development]

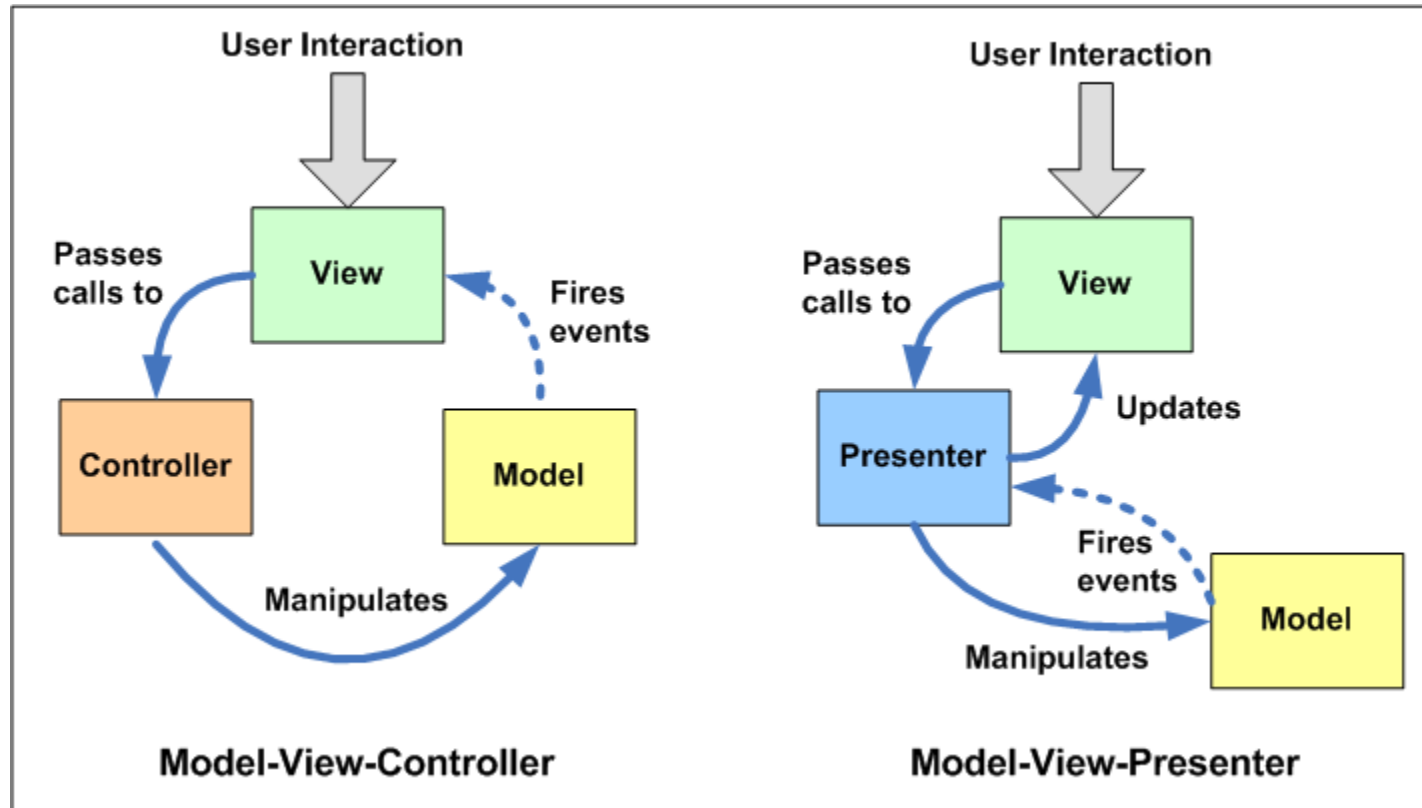
- Application framework
  - iOS sandboxes an app as a traditional application on an OS (MVC)
  - Android model an app as a set of Services, activities & Content Providers, MVP realized in Adaptor-backed Views



# Objects in iOS App



# Or MVP? MVVM?



Stack Overflow: MVC vs MVP

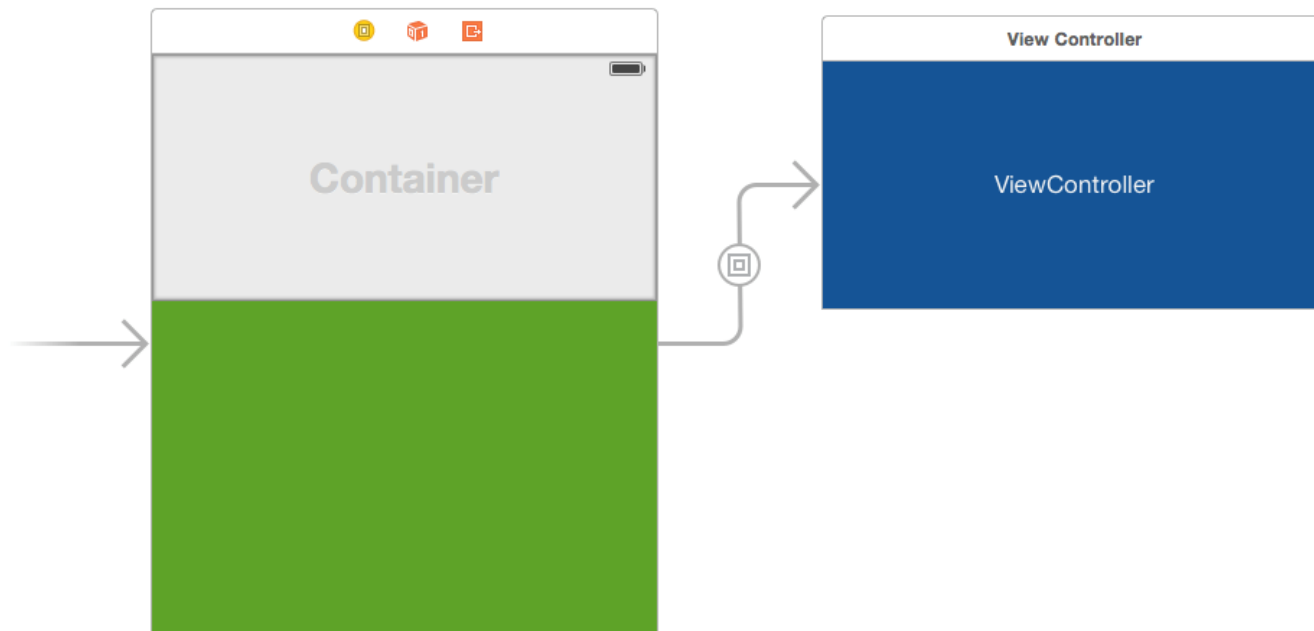
# RecyclerView in iOS?

- Similar Adapter-ViewHolder design pattern is also used in iOS:

Android	iOS
RecyclerView	UICollectionView
Adapter	UICollectionViewDataSource
ViewHolder	UICollectionViewCell
LayoutManager	UICollectionViewLayout

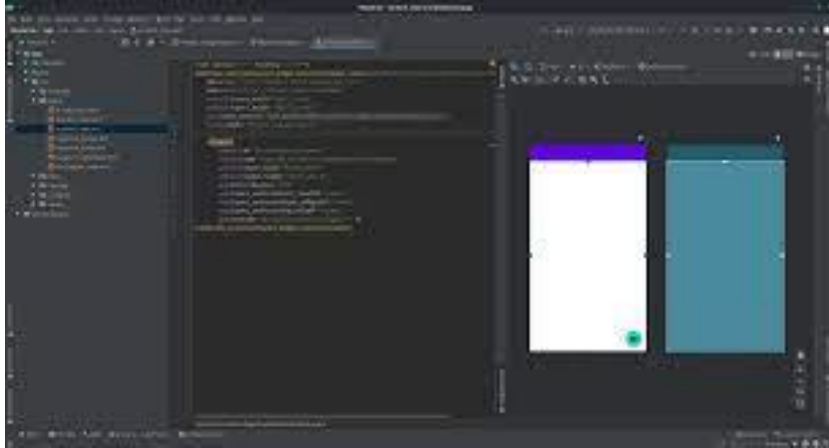
# Fragment in iOS?

- iOS can do something similar by include child view controller inside another view controller



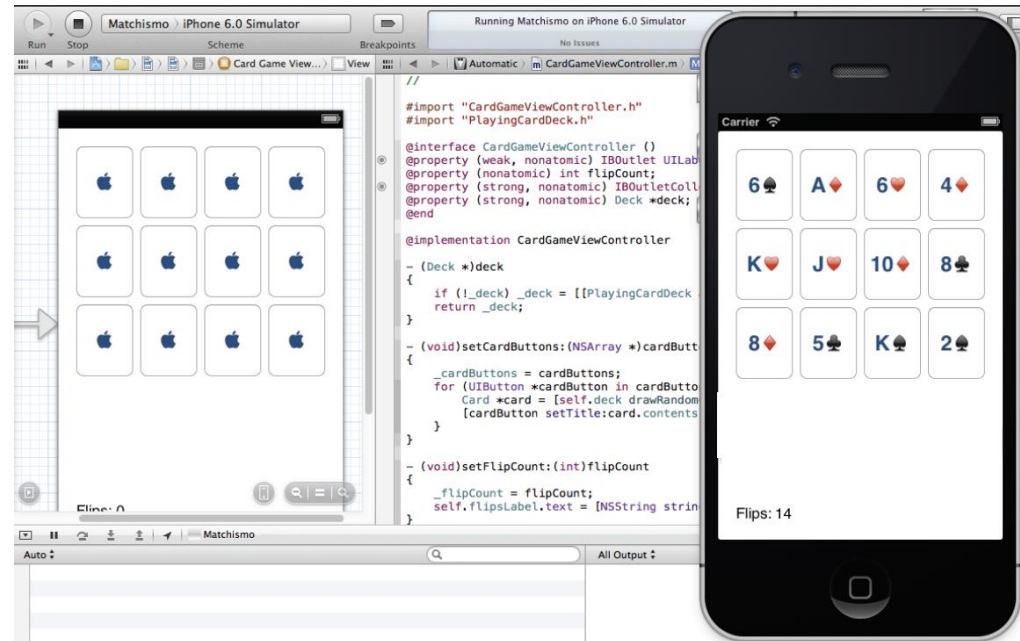
# Android vs. iOS [IDE UI]

- IDE User interface
  - Xcode is very user friendly
  - Android Studio is having significant improvement in IDE UI



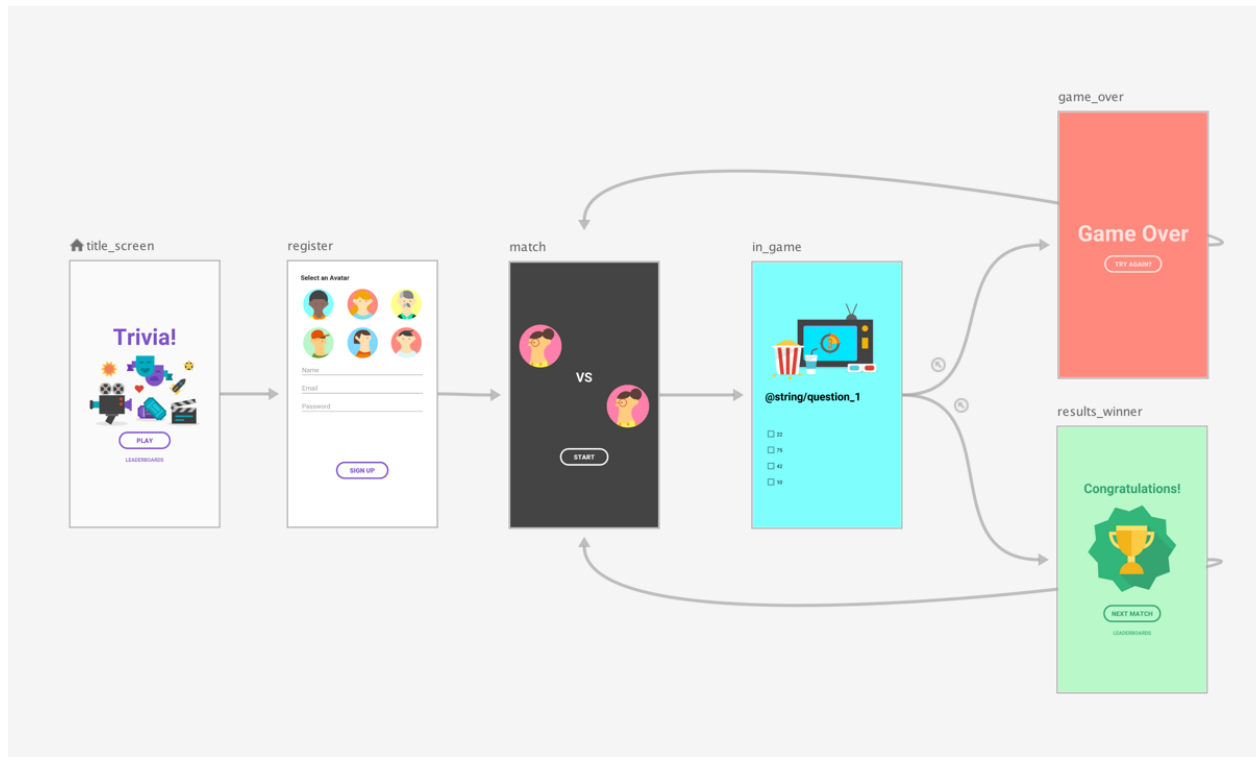
# Xcode & iOS Storyboard

- graphically construct and configure your application's windows and views
- Starting from iOS5, storyboard was introduced to enable pure graphical way for editing MVC's view

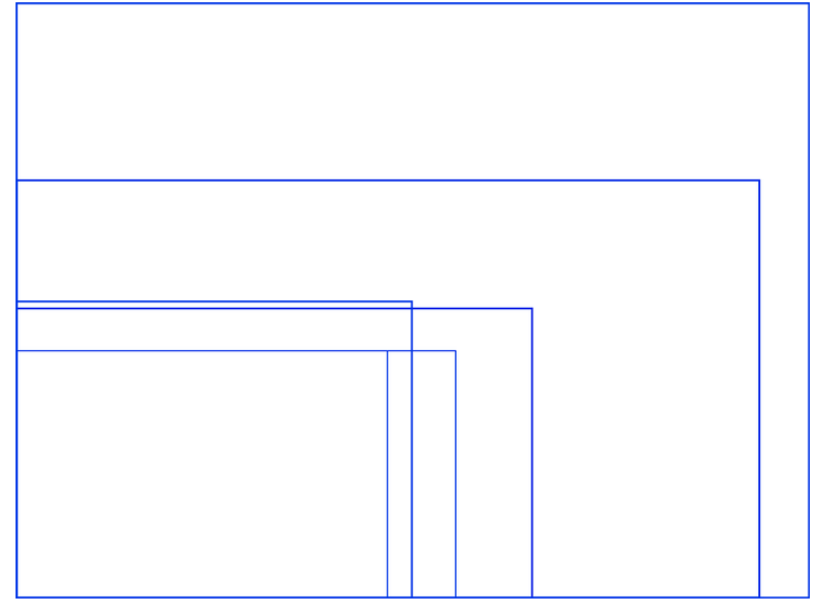
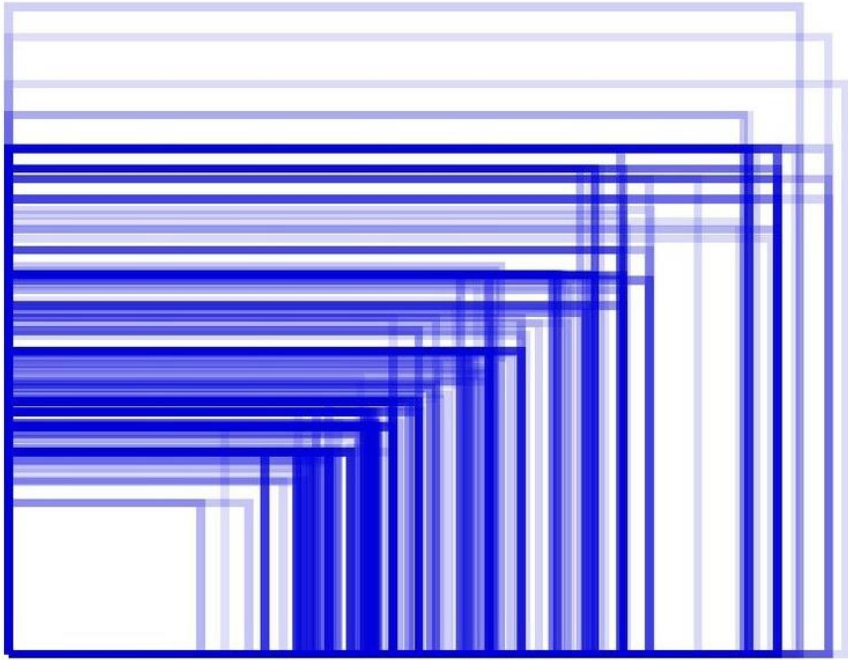


# Android Jetpack Navigation

- Android introduces similar Navigation Editor via Android Jetpack Library since Android Studio 3.2



# Android vs. iOS [Screen fragmentation]



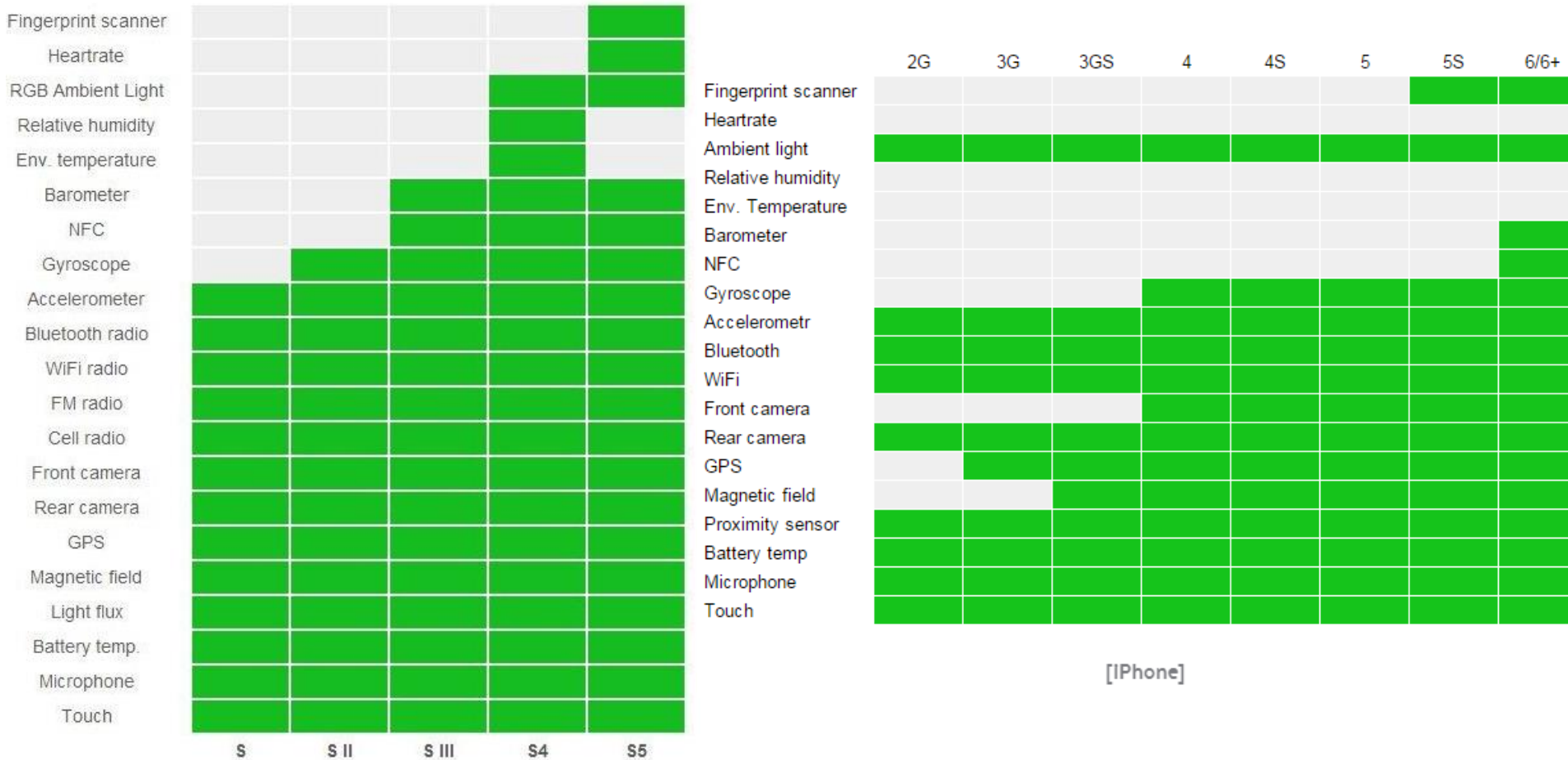
Possible screen size in Android & iOS

The current [Android SDK](#) version supports **more than 20 resolutions**. However, only about **ten** of these are in active use.

Apple, on the other hand, has only **six generalized sizes** for its screens. The only big change that happened with Apple devices took place last year when they moved to Retina displays – the resolution doubled, while screen sizes remained the same.



# Android vs. iOS [Hardware Components]



Mostly similar in both platforms but sensors on different devices vary in terms of **precision tracking**, motion **processing**, measurement **ranges**, and so on, which means that an **app may perform differently** depending on the mobile device it's running on.

# Android vs. iOS [support & revenues]

- Documentation & Support

- Both iOS & Android now have extensive documentation & sample code, whereas Android backward support need more works (iOS basically inhibit old versions); *StackOverflow* gives non-official ample solutions
- iOS is simpler and coherent
- Openness of Android brought difficulties in supporting huge variety of hardware spectrum, which is problematic to small developers; New layout features did help to ease the pain

- Revenues

- Apple's In-App Purchase brings more flexibility to app users
- Android is catching up with iAd, In-App Purchase.

# Android vs. iOS [Optimization]

- Performance
  - iOS has advantage of C's high performance
  - To combat the efficiency issues with Java Virtual machine, ART is developed, NDK can also be used

# Android vs. iOS [Optimization]

- Hardware-aware Optimization
  - iOS's emulator is based on x86-64 Mac hardware, so ARM aware optimization cannot be tested on emulator
  - Android Studio provides ARM emulator images for ARM intrinsic (C-like wrapper of assembly) for initial optimization tests.

# Reference

- Android vs iOS

[https://www.diffen.com/difference/Android\\_vs\\_iOS](https://www.diffen.com/difference/Android_vs_iOS)

- Does iOS have something similar to Android's RecyclerView?

<https://stackoverflow.com/questions/34227364/does-ios-have-something-similar-to-androids-recyclerview>

- Equivalent of Fragment (Android) in iOS

<https://stackoverflow.com/questions/28167644/equivalent-of-fragment-android-in-ios>