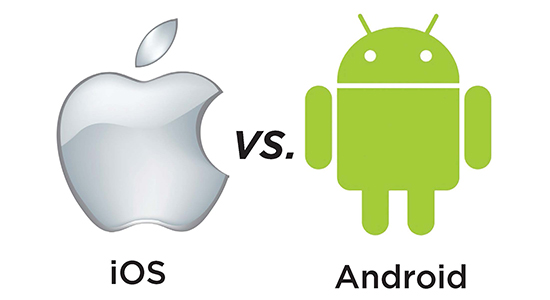
When deciding to platform you want to develop for a few considerations come to mind. How big is my user-base? What do development costs look like? Is the programming language complex? How easy are updates or patches with each platform?

We can look objectively and the pros and cons of both platforms to determine which may be more appropriate for the application we are developing. Android developers utilize Android Studio, a proprietary tool introduced by Google in 2013 and sporting an expanded range of available features. This integrated development environment has cross-platform support, high readability, a wide range of development and debugging features. iOS developers, on the other hand, rely on the proprietary XCode tool. The Apple-backed solution provides a variety of bug fixing tools, supports the entire range of iOS devices, and is easy to get around.



# Market share

According to [StatCounter](https://gs.statcounter.com/os-market-share/mobile/), Android hold most of the market share. Where iOS holds only 16% of worldwide smartphone users, Android devices are popular among over 80% of the population. When it comes to market share by regions, here’s what the statistics have to say:

US - 52.59% - iOS | 46.98% - Android

Europe - 22.23% - iOS | 72.71% - Android

Asia - 13.85% - iOS | 83.09% - Android

|  |  |  |  |
| --- | --- | --- | --- |
|  | Development | Platforms |  |
| Android | Platform | iOS | Platform |
| Pros | Cons | Pros | Cons |
| The Android market has a broad global reach and has a lot of earning potential. | Android may take longer to develop for because there are so many different Android OS versions on the market. Different devices get important security updates at different times, and this makes maintaining and developing Android apps more challenging. | Due to the complex process of application uploading and moderation, App Store is a safer source of apps than Google Play. There’s less risk to get tricked by a malware distributor | Apple takes a 30 percent cut from app developers who make over $1 million through the ‌App Store‌ on an annual basis, but for small developers who make less, Apple has cut its fees to 15 percent through the Small Business Developer Program. |
| Android is more customizable than iOS | Sometimes it consumes too much memory | The Apple-backed solution provides a variety of bug fixing tools, supports the entire range of iOS devices, and is easy to get around. | Provides limited widget support over Android. |
| Android Studio helps quickly generate multiple versions of your app for various devices and offers a flexible Gradle-based build system. | Google has charged a 30 percent cut for any purchases through the Google Play Store since it first launched as the “Android Market” | Xcode has a decent source editor and assistant editor. | Android app marketplaces have surpassed iOS by sales |
| Advanced code editor and layout designer | App security isn’t as stringent on the google play store | iOS apps prove to have a higher return on investment than Android apps | App Store is stricter in terms of app moderation - thus, there’s a risk that your app will never be published. |

# Developing my App on iOS

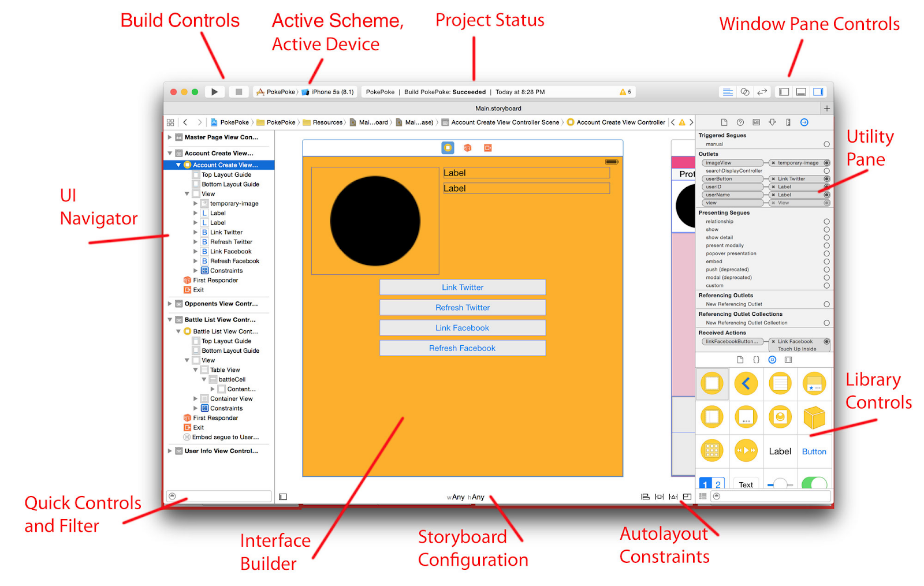
Developing my application for iOS wouldn’t be all too different than Android studio. The basis of the application involves buttons that bring the user to the main functions of the application. The user inputs their character data and the app saves the data to be called upon at a later time. The dice roller function is a simple random number generator which changes depending on the dice face the user chooses. In the following section we’ll review how iOS implements these functions within XCode.

# XCode Functionality

Source: <https://developer.apple.com/documentation/>

Widgets: Open your app project in Xcode and choose File > New > Target. From the Application Extension group, select Widget Extension, and then click Next. Enter the name of your extension. If the widget provides user-configurable properties, check the Include Configuration Intent checkbox.

Layouts: To create constraints select the button and click the Align icon in the auto layout menu. A popover menu will appear, check both “Horizontal in container” and “Vertically in container” options to center the button on the screen. Then click the “Add 2 Constraints” button.



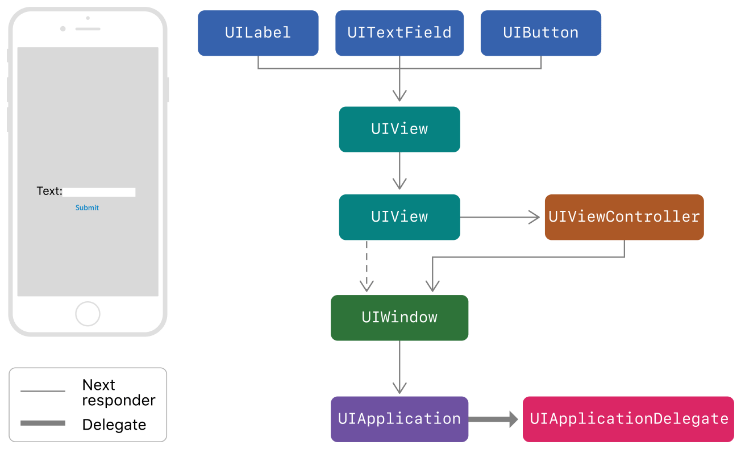
Buttons: Access the object Library by:

* selection a View in Storyboard
* clicking on + Button at top right of Xcode window
* Button(s) will be listed here: Graphical user interface, text, application, email

  Description automatically generated

XCode event handling:  
Apps receive and handle events using responder objects. A responder object is any instance of the UIResponder class, and common subclasses include UIView, UIViewController, and UIApplication. Responders receive the raw event data and must either handle the event or forward it to another responder object. When your app receives an event, UIKit automatically directs that event to the most appropriate responder object, known as the first responder.

Unhandled events are passed from responder to responder in the active responder chain, which is the dynamic configuration of your app’s responder objects. Figure 1 shows the responders in an app whose interface contains a label, a text field, a button, and two background views. The diagram also shows how events move from one responder to the next, following the responder chain.



# References and Sources

<https://developer.apple.com/documentation/>

<https://www.wisbar.org/NewsPublications/WisconsinLawyer/Pages/Article.aspx?Volume=90&Issue=5&ArticleID=25605>

<https://developer.apple.com/forums/thread/693758>

<https://developer.apple.com/documentation/widgetkit/creating-a-widget-extension>

<https://www.ego-cms.com/post/5-major-differences-between-ios-and-android-app-development>

<https://www.mooc.org/blog/ios-vs.-android-app-development#:~:text=For%20both%20new%20and%20experienced,use%20Java%20and%2For%20Kotlin>.