

スマホプログラミング

編集にあたって

塚本昌彦 | 神戸大学 稲見昌彦 | 東京大学

スマートフォンはここ10年の間に急激に浸透し、台数ベースでパソコンをはるかに追い抜くほどになった。今や人類のコンピュータプラットフォームとして最もポピュラーなものになったといえる。研究開発分野におけるアプリケーション開発においてもスマートフォンをターゲットとすることが多くなってきており、コンピュータに携わる多くの人がスマートフォンプログラミングをマスターしたいと思うようになってきたのではないだろうか。一方で、スマートフォンプログラミングが大学等の教育機関で教えられることは少なく、その必要に駆られた人は多くの場合書籍やWebなどを使って自力で学んでいるというのが現状である。

スマートフォンプログラミングを学ぶ1つの難しさは技術の移り変わりの速さにある。本誌でも、52巻4・5号(2011年4・5月号)で新年度・ゴールデンウィークチュートリアル「スマートフォンプログラミング～iPhoneとAndroid」、53巻1号(2012年1月号)で冬休みチュートリアル「スマートフォンプログラミング2～iPhone/Android/Windows Phoneによるセンサプログラミング～」を特集し、おそらくそのときにそれを参考にプログラミングしてみたという人も少なからずいたのではないかと思

われる。それから5年以上たってスマートフォンを取り巻く環境、特に開発環境は大きく変わっているといえる。特にこの5年間は、Swift、Android Studio、Unityという3つの技術やシステムの登場と台頭という3つの大きな変革があった点が重要である。また、バージョンやURLなど細かい部分も変化しており、5年前の記事はすでに古いものになっている。記事の「賞味期限」は図らずも短期間で切れてしまったのである。Webで入門ページを探した場合も、それがどれくらい最新のものか分からない場合があり、最新のものでない場合には勉強を進める上でいろいろと行き詰まる原因となり得る。それがまさに今回改めてスマートフォンプログラミングの入門編を再特集した理由でもある。

本小特集では、前述の3つの技術、システムを取り上げる。周知の通り、現状のスマートフォンはiOSを搭載するiPhoneとAndroidを搭載するものの2つが主流であり、これらで動作するアプリケーションを開発するメジャーな方法をカバーする。

最初の解説は、Swiftを用いたiPhone向けのアプリケーション開発環境である。前回、前々回は言語としてはObjective-Cを取り上げたが、その後Swiftが発表され、そちらの方がメジャーになりつ



つある。これも同じく前回、前々回と同じ執筆者である沼田氏に、Swift 言語自体を含めてやや詳しく解説をお願いした。

2 番目の解説は、Android 向けのアプリケーション開発環境である Android Studio を用いた開発手法である。前回、前々回の特集では主に Eclipse と Java を使ったものを取り上げたが、その後 Android Studio が登場し、専用の開発環境としてインストールから実行まで、おそらくこちらのほうが手軽に行えるようになった。前回、前々回の特集で解説を行っていただいた石丸氏に今回も解説をお願いした。開発言語は前回、前々回と同じく Java である。

3 番目の解説が、最近特にパソコンを含めてマルチプラットフォームで高性能なアプリケーションを開発するのに有用とされる Unity である。Unity は、執筆者の築瀬氏が述べている通り、スマホアプリ開発においても広く利用されるようになってきており、この領域で重要性が高いものと考えられる。iOS、Android 問わずアプリケーションが開発できる点も大きなメリットである。

さて、これら以外にもスマホ向けのアプリケーション開発の方法には下記のようなものが存在する。

- Web アプリとして開発するもの。HTML5 や JavaScript を用いてアプリケーションを開発し、スマホ上のブラウザでアプリ実行するものである。
- Web アプリとネイティブアプリのハイブリッド型のもの。スマホのデスクトップ画面からネイティブアプリを起動し、その中からインターネットに接続して Web アプリを実行するというものである。

一方で開発環境としては、以下のようなものが挙げられる。

- iOS, Android それぞれに対する開発環境がいくつかある。Android 向けの Processing など。
- Unity 以外の統合開発環境もいくつかある。Visual Studio や IntelliJ IDEA など。
- Xamarin, Apache Cordova などは開発フレームワークである。
- スマホ上でプログラミングして実行できるようなツールもある。Scratch, MIT App Inventor など。これらは主に学習向けで、実践的なアプリケーションを提供するものではないことが多い。

さらに進んだスマホアプリの形態として、クラウド連携や IoT デバイス連携、AI や学習も最近のトレンドであり、それに向けた開発環境、プログラム言語も存在する。特に JavaScript は多くのプラットフォームに対応しており、単一の言語でサーバからエッジ、IoT までのプログラムの連携が記述できるというメリットがある。ロボットやドローンの制御をスマホ上のアプリでプログラミングするようなケースも多い。紙面の制限のため、本小特集ではこのような発展的な内容にまで踏み込まず、初歩的な部分のみを対象とした。より進んだプログラミングを学びたい人のためへの参考書や参考 Web ページなどは各章で示しているので、ご希望の方はぜひより深いスマホプログラミングの世界へと進んでいただきたい。また先にも述べた通り、今回の記述内容も月日が経つにつれ古くなっていくことが考えられる。本誌発行時から年月を経て本小特集を参照する場合にはご注意ください。

(2017 年 12 月 8 日)