iOS SDK (第1回)

～iOSの概要とiOS SDKの導入方法～

my recommendations on research & development tools.

松村将太郎[†], (正会員) 熊澤逸夫[†]

キーワード：モバイルアプリケーション開発, iOS, iOS SDK, Xcode

1. ま え が き

近年, iPhoneやAndroid携帯をはじめとするスマートフォンが急速に普及し, パソコン市場においてもタブレット型PCのシェアが拡大している. こうしたスマートフォンやタブレット型PCの特色は, 一般ユーザがアプリを独自に開発し, 全世界に向けて発進, 販売する仕組みが提供されていることである. また, これらの機器にはタッチインタフェースや加速度センサ, カメラといった多くのセンサ機能が標準で搭載されており, これらの機能を単独または組合せて使うことにより, 個性的なアプリケーションを作ることができる. さらに最近では, 内部プロセッサの強化により, 画像処理などの計算量の大きい処理も行えるようになってきた.

今回の「私の研究開発ツール」では, 5回にわたって, iPhoneやiPod touch, iPadなどのiOSを搭載するモバイル機器で, アプリケーションの制作を行うため方法について紹介する. 第1回目は, これらのモバイル機器に搭載されているiOSの概要と, iOS専用のアプリケーションを制作するためのツールであるiOS SDKの導入方法について解説する.

2. iOSとは

iOSは, スマートフォンのiPhoneやメディアプレーヤのiPod touch, タブレットのiPadなどに搭載されているOSである. iOSは従来, iPhone OSと呼ばれていたが, iPadとOSが共通化されたことによりiOSに変更され, 現在バージョン5までリリースされている.

iOSは図1に示すように, Core OS, Core Services, Media, Cocoa Touchの四つの階層から構成されている. 下の階層はOSの基本的な機能が含まれており, 上の階層にいくほど下の階層の機能が抽象化された機能が含まれている. 各階層の機能はC言語またはObjective-C言語で実装されており, 下の階層にいくほどC言語で実装されている場合が多く, 上の階層にいくほどObjective-C言語で実

図1 iOSの階層構造

表1 各階層の機能

Core OS	OSの中核を成す最も下位の階層であり, メモリー, スレッド, プロセス, ファイル, ネットワークといったOSの基本機能を提供する. すべての機能はC言語APIとして提供される.
Core Services	アプリケーションを構築するうえで最も基本的な機能を提供する階層であり, 文字列, 配列・集合といったコレクション型データ, 電話関連の情報や位置情報などの機能を提供する. この階層の機能はC言語APIまたはObjective-C APIで提供される.
Media	グラフィック, サウンド, ビデオといったマルチメディア機能を提供する階層である. この階層の機能はC言語APIまたはObjective-C APIで提供される.
Cocoa Touch	Cocoa Touchレイヤは, 最も上位の階層であり, アプリケーションを構築するうえで重要なユーザインタフェースの機能を提供する. また, ネットワーク接続やセンサ情報などの下位階層の機能を高度に抽象化した機能も提供する. すべての機能はObjective-C APIで提供される.

装された機能が多く存在する. 各階層の説明を表1に示す.

各階層には, Frameworkと呼ばれるある決まった手続きによって機能(メソッドや関数)を呼び出すことで, 作業を達成できる枠組みが用意されている. 下記に, 各階層に含まれる代表的なFrameworkを簡単に紹介する.

(1) Core OSレイヤの代表的なFramework

- (a) Accelerateフレームワーク: 線形代数などの数値計算を行うLAPACKおよびBLASライブラリや, 音声・画像などの信号処理において高パフォーマンスのベクトル計算を行える機能を提供する. (C言語API)

[†] 東京工業大学 像情報工学研究所

"iOS SDK (1): Outline of iOS and how to install iOS SDK" by Shotaro Matsumura and Itsuo Kumazawa (Imaging Science and Engineering Laboratory Tokyo Institute of Technology, Yokohama)

(2) Core Servicesレイヤの代表的なFramework

(a) Core Foundationフレームワーク：文字列、配列・集合などのコレクションデータ、日付などのアプリケーションを構築するための基本的な機能を提供する(C言語API)

(b) Foundationフレームワーク：C言語で記述されたCore Foundationフレームワークの機能をObjective-Cでアクセスするためのラッパーを提供する(Objective-C API)

(3) Mediaレイヤの代表的なFramework

(a) AV Foundationフレームワーク：オーディオファイルの再生やマイクやカメラなどからのメディアキャプチャを行う機能を提供する(Objective-C API)

(b) Core Graphicsフレームワーク：2Dグラフィックス描画エンジンであるQuartzを扱うための機能を提供する(C言語API)

(c) Audio Toolboxフレームワーク：システム音の再生やオーディオファイルの読み書きや変換などの機能を提供する(C言語API)

(d) Audio Unitフレームワーク：音声信号処理モジュールのプラグイン規格であるAudio Unitを扱うための機能を提供する(C言語API)

(e) OpenGL ESフレームワーク：3Dグラフィックする描画を行うためのライブラリーであるOpenGL ES 1.1およびOpenGL ES 2.0を利用するための機能を提供する(C言語API)

(4) Cocoa Touchレイヤの代表的なFramework

(a) UIKitフレームワーク：ボタンやビューなどのグラフィカルなユーザインタフェースやアプリケーションの管理、タッチイベント処理、加速度センサやカメラへの簡易なアクセス方法などのインタフェースに関する多くの機能を提供する(Objective-C API)

3. iOS SDKを導入する前に

iOS SDKを導入する前に、開発に必要な機材とiOS SDKを入手するために必要なApple Developerへの登録および実機でアプリケーションを動かすために必要なiOS Developer Programへの加入について紹介する。

3.1 開発に必要な機材

開発を行うためには、Mac OS Xを搭載したIntel CPU搭載のMacintosh(Mac mini, MacBook, Mac Proなど)が必要で、開発したアプリを実機で動かす場合は、iOSを搭載したデバイスが別途必要となる。

また、執筆時点で最新のSDKであるiOS 5 SDKをMacintoshに導入して開発を行うためには、MacintoshのOSであるMac OS Xのバージョンを10.7(Lion)以上

にしておく必要があり、一つ前のバージョン10.6(snow Leopard)では、iOS 4 SDKまでしか導入できないため注意する必要がある。

3.2 Apple Developerへの登録

iOS SDKを入手するためには、Apple Developerへの登録が必要となる。Apple Developerへの登録はiOS Dev Center¹⁾から行うことができる。Apple Developerへの登録には別途Apple IDが必要となるが、持っていない場合は新規登録をその場で行うことができる。

3.3 iOS Developer Programへの加入

iOS SDKはApple Developerへの登録を行うことで無料で入手可能であるが、作成したアプリケーションを実機で動かすことはできない。作成したアプリケーションを実機で動かすためには、別途iOS Developer Programへの加入が必要となる。

iOS Developer Programには個人、法人、企業、大学教育向けの4種類が存在する(2011年11月30日現在)。

(1)個人向け-年間8,400円

(2)法人向け-年間8,400円

(3)企業向け-年間24,800円

(4)大学教育向け-無料

各プログラムの年間の費用以外の違いについてはiOS Developer Programの比較のサイト²⁾を参照のこと。

4. iOS SDKの導入

iOS SDKを導入するためには、iOS Dev CenterからXcodeをダウンロードしインストールすることで、iOS SDKも一緒にインストールされる。

Xcodeとは、MacやiOS用のソフトウェアを開発するための統合開発環境である。Xcodeには、バージョン3のXcode3とバージョン4のXcode4の二つがあり、両方で操作方法が大きく異なっている。ただし、Xcode3はiOS 5 SDKには対応しておらず、最新のXcode4はMac OS X 10.7(Lion)以上にしか対応していない。本稿では、Xcode4を使ったアプリケーションの開発方法について説明する。

XcodeをインストールするとiOS SDKの他に、下記の機能も一緒にインストールされる。

・iOS Simulator

・Instruments

iOS Simulatorはその名の通り、PC上でiOSアプリケーションのシミュレーションを行うことができるツールであり、iPhoneとiPad用が存在する。Instrumentsは、Xcodeで作成したアプリケーションのパフォーマンスなどを解析することができるツールであり、CPU・GPUの付加やメモリーリーク等をリアルタイムで解析することができる便利なツールである。

4.1 iOS SDKのインストール

- (1) iOS Dev Center¹⁾へログインし、トップページの「Downloads」からXcode4をダウンロードする。
- (2) ダウンロードしたXcode4のディスクイメージ(dmgファイル)をマウントし、手続きに沿ってインストールを行う。

4.2 iOS Simulatorでサンプルを動かす

- (1) Xcodeを起動し、上部メニューバーから「File」⇒「New」⇒「New Project...」を選択する。
- (2) iOS Applicationの「OpenGL Game」を選択し、「Next」をクリック。
- (3) プロジェクトの名前を入力し、「Next」をクリック。
- (4) プロジェクトの保存場所を設定し、「Create」をクリックするとプロジェクトが作成される(図2)。
- (5) ツールバーの「Scheme」を「iPhone Simulator」に変更する(図3)。
- (6) 「Run」ボタンをクリックするとプログラムがビルドされ、iOS Simulatorが立ち上がる(図4)。

5. デバイスでアプリを実行する

4.2節のサンプルをデバイスで動作させるためには、iOS Provisioning Portalにおいて「Certificates」、「Devices」、「App IDs」、「Provisioning」の四つの項目について順に設定を行い、Provisioningファイルを取得する必要がある。

ただし、iOS Provisioning Portalでの作業方法については、守秘義務のため詳しく説明できない。そのため、iOS Provisioning Portalにおける設定は、サイトの説明をよく読んで進めていただきたい。

ここでは、iOS Developer Programに加入済みとして話を進める。

5.1 証明書の取得

- (1) Finderを開き、「アプリケーション」⇒「ユーティリティ」⇒「キーチェーンアクセス」の順でキーチェーンアクセスを開く。
- (2) 上部メニューバーから、「キーチェーンアクセス」⇒「証明書アシスタント」⇒「認証局に証明書を要求」を開く。
- (3) 証明書情報にiOS Dev Centerに登録したメールアドレスと名前を入力し、「ディスクに保存」と「鍵ペア情報を指定」にチェックが入った状態で「続ける」をクリック。
- (4) ファイルの保存場所を指定し、「保存」をクリック
- (5) 鍵ペア情報の鍵のサイズが「2048ビット」、アルゴリズムが「RSA」となっていることを確認し「続ける」をクリックすると指定した保存場所に証明書を要求するファイルが作られる。
- (6) iOS Dev Centerへログイン後、iOS Provisioning Portalに移動し、(5)で作成した証明書を要求するファイルを使って証明書を作成する。
- (7) 作成した証明書をダウンロードし、ダウンロードした証明書をダブルクリックすると証明書がキーチェーンに登録される。登録されたかを確認するには、キーチェーンアクセス画面左メニューの「証明書」を選択する。

5.2 デバイスの登録

- (1) 開発で用いるiOS搭載デバイスをUSBケーブルを使

って、Macとつなぐ。

- (2) Xcodeを起動し、ツールバー右端の「Organizer」ボタンをクリックするとOrganizerが立ち上がる。
- (3) 左のメニューから接続されているデバイスを選択し、画面のデバイス情報のIdentifierにある40桁の英数字 (Device ID) を確認する (図5)。「Use for Development」ボタンが表示されている場合は、ボタンを押して設定を行うと、デバイスの情報が表示される。
- (4) iOS Provisioning Portalにおいて、(3)で確認したDevice IDを使い、開発用のデバイスを登録する。

5.3 App IDsの登録

iOS Provisioning Portalにおいて、App IDを作成する。App ID (Bundle Identifier) は、アプリケーションを識別するためのIDであり、異なるアプリケーションごとに一つのIDを割当てることになる。一度登録したApp IDは削除することができないので注意する必要がある。

5.4 Provisioningファイルの取得

- (1) iOS Provisioning Portalにおいて、これまでに設定した情報を使ってProvisioningファイルを作成する。
- (2) 作成したProvisioningファイルをダウンロードする。
- (3) ダウンロードしたProvisioningファイルをダブルクリックすると接続されているデバイスにインストールされる。デバイスにインストールされたかどうかについては、Organizerのデバイス情報のProvisioningで確認できる (図5)。

5.1節から5.4節までの設定を行うと、Organizerの接続されているデバイス名の隣の○が緑色になり、そのデバイスを使用しての開発が可能となる。緑色になっていない場合は、どこかの設定が間違っている可能性があるため、表示されている警告を見て対処していただきたい。

5.5 デバイスでサンプルを動かす

4.2節で作成したプロジェクトを開き、ツールバーのSchemeの設定を「iPhone Simulator」から「開発用のデバイス名」に変更する。最後に「Run」ボタンを押すと、iOSデバイス用にプログラムがビルド・インストールされ、デバイス上でプログラムが実行される。

ただし、デバイスのiOSのバージョンが最新でない場合は、プログラムをインストールできない可能性があるため、そのような場合は、編集画面の「Deployment Target」をデバイスのiOSのバージョン以下に設定する必要がある。

6. む す び

本稿では、iOSの概要について説明し、iOS SDKを導入してから実機でサンプルプログラムを動作されるまでの方法を紹介した。次回以降は、概要で紹介したFrameworkを使って、実際にXcodeを使ってiOSのプログラミングを行う方法について解説していく予定である。 (2011年11月28日受付)

図5 Organizer画面

図6 iOSデバイス (iPod Touch iOS v5.01) で実行

〔文 献〕

- 1) <http://developer.apple.com/devcenter/ios/index.action>
- 2) <http://developer.apple.com/jp/programs/start/ios/>
- 3) Apple Inc.: “iOSテクノロジーの概要”, <http://developer.apple.com/jp/devcenter/ios/library/documentation/iPhoneOSTechOverview.pdf>
- 4) 柴田文彦: “Xcode4入門for iOS/Mac OS X”, 秀和システム
- 5) 國居貴浩: “iOSデバッグ&最適化技法”, 秀和システム

松村将太郎 2011年、東京工業大学大学院総合理工学研究科博士後期課程修了。同年、同大学像情報工学研究所研究員。ユーザインタフェースや画像処理などの研究に従事。博士 (工学)。

熊澤 逸夫 1981年、東京工業大学工学部電気電子学科卒業。1986年、同大学大学院博士課程修了。同年、同大学工学部助手。現在、同大学像情報工学研究施設教授。パターン認識・画像処理、人工ニューラルネットワーク、認知知覚モデルの研究に従事。博士 (工学)、正会員。