



iPhone アプリ開発中級教育に関する一考察

北 門 達 男

1. はじめに

現在、我々は ICT 業界が25年ぶりに迎えた大きなパラダイムシフトの真只中にある。すなわち、スマートフォン/タブレット端末とクラウドコンピューティングサービスが ICT 業界を大きく変えつつある。ノキア、マイクロソフト、HP など数年前の ICT 業界の覇者が新しい波に乗り遅れる現象も現れ始めている。

この新しい波に対応するため、2009年から「iPhone アプリ開発コース」の開発を行ってきた。2012年8月には、38号館に70台の iMac が導入され、開発環境も充実してきている。2012年前期から「iPhone アプリ開発入門」の教育をゼミ生を対象に開始し、後期から IT ビジネス特修実習として正式に教育を開始した。2013年前期終了時点で、23名がこのコースを受講した。受講アンケート回答では、「半期のコースを1年間のコースに拡大してほしい」「Apple 社の提供するフレームワークの中身がよくわからない」といった意見が大半を占めた。そこで、従来から研究してきた中級コースの内容について述べたい。

2. MVC モデル対応開発とプログラミング主体開発の比較

iPhone アプリの開発環境である Xcode のフレームワークはオブジェクト指向開発で用いられる MVC モデルに適応しているが、各種の制約条件がある。例えば、データベースアプリを開発するには、CoreData が利用できるマルチビューのアプリしか簡単に作成することができない。シングルビューのデータベースアプリを開発するには SQLite を直接利用するのが適している。このように、自由度の高い独自のアプリを開発しようとする、Apple の提供するフレームワークを使用せずに、プログラミングを主体とした開発に移行せざるを得なくなる。表1にこれらの開発方法の特徴を比較した。

表1 MVCモデル主体開発とプログラミング主体開発の比較

	MVCモデル主体開発	プログラミング主体開発
適用分野	入門, RAD 開発に向いている。	中級から上級に向いている。部品化などは独自に構築しなければならない。
Model	CoreData と Xcode のモデラーを利用することができる。	SQLite に直接アクセスする。データベース構築ツールを独自に組合せなければならない。
View	GUI 構築ツールである Story Board を利用できる。	プログラムで直接 UIKit や各種ライブラリーを利用する。
Control	Xcode のテンプレートを最大限活用する。	プログラミング主体となる。独自の部品化を行わなければならない。
長所	短期間で iPhone の標準 GUI に対応したアプリを開発することができる。	iPhone のテンプレートに制約されない独創的なアプリを開発することができる。
短所	画一的なアプリしか開発できない。	iOS や UIKit, データベースアクセスなどに対する高い技術力が要求される。

3. SQLite を直接操作するデータベースアプリケーション例

(1) SQLite を直接操作するデータベースアプリケーションの概要

iPhone アプリ開発教育を行う上で、教員側の説明もしくは受講者の理解という面で苦勞するのが CoreData を用いたデータベース・アプリである。CoreData の使える条件は限定されており、自由度の高いデータベース・アプリを開発するには、SQLite を直接操作の方が望ましい場合が多い。そこで、中級コースで早期にこの方法を学習することが、その後のスキルアップに役立つといえる。図1にその簡単な例を示す。


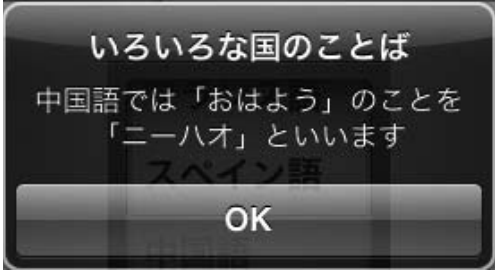
画面例	説明
	<ul style="list-style-type: none"> • このアプリは SQLite を直接使用する例である。 • 上のセグメントコントロールでフレーズを選択する。 • 下のピッカーで言語を選択する。 • 「ことばの表示」ボタンを選択すると、下図のようにその言語でどう言うかを表示する。 

図1 SQLite を直接操作するデータベースアプリケーション例

(2) データベースの準備

SQLite を直接利用するには、Xcode のモデラーを利用することができない。そこで、このアプリのデータベースの元となる以下のような Phrases.csv を準備する。

1. ‘英語’, ‘ハロー’, ‘サンキュー ベリーマッチ’, ‘グッバイ’
2. ‘ドイツ語’, ‘ゲーテンターク’, ‘ダンケ シェーン’, ‘アウフビーダーゼーエン’
3. ‘ロシア語’, ‘ズドラーストヴィーチェ’, ‘バリショエ スパシーバ’, ‘ダスヴィダーニャ’
4. ‘フランス語’, ‘ボンジュール’, ‘メルシー ボクー’, ‘オ ルヴォワール’
5. ‘スペイン語’, ‘オーラ’, ‘ムチスィマス グラシアス’, ‘アディオス’
6. ‘中国語’, ‘ニーハオ’, ‘シェイ シェイ (謝謝)’, ‘ザイ チェン (再会)’
7. ‘韓国語’, ‘アンニョン (安寧) ハシムニカ’, ‘カムサ (感謝) ハムニダ’, ‘アンニョンヒ カセヨ (送る人) / アンニョンヒ ケセヨ (去る人)’

このファイルをもとにSQLite データベースを作成する。FireFox のアドオンツールに SQLite Manager があり、SQLite データベースを作成・管理することができる。

(3) データベースの作成

SQLite Manager を用いて Phrases.csv から Phrases.sqlite データベースを作成してプロジェクトに追加する。図 2 にこのアプリ用の SQLite Manager 画面例を示す。

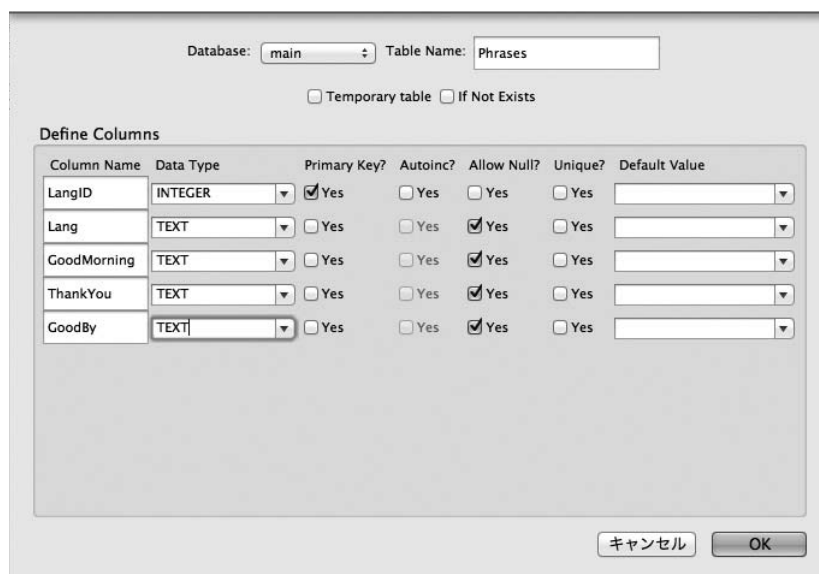


図 2 SQLite Manager 画面例

(4) データベースラッパークラスの作成

SQLite を直接利用するためにデータベースラッパークラスを作成する。以下のリストは、その中のメソッドである。このクラスが完成すると、後は容易に利用することが可能となる。

```

-(NSMutableArray*)getAllPhrases{
    NSMutableArray *phrases = [[NSMutableArray alloc] init];
    const char *sql = "SELECT * FROM Phrases";
    sqlite3_stmt *statement;
    int sqlResult = sqlite3_prepare_v2(database, sql, -1, &statement, NULL);
    if (sqlResult==SQLITE_OK) {
        while (sqlite3_step(statement) == SQLITE_ROW) {
            Phrase* phrase = [[Phrase alloc] init];
            char* lang = (char*)sqlite3_column_text(statement, 1);
            char* goodmoring = (char*)sqlite3_column_text(statement, 2);
            char* thankyou = (char*)sqlite3_column_text(statement, 3);
            char* goodbye = (char*)sqlite3_column_text(statement, 4);
            phrase.LangID = sqlite3_column_int(statement, 0);
            phrase.Lang = (lang) ? [NSString stringWithUTF8String:lang] : @"";
            phrase.GoodMorning=(goodmoring)? [NSString stringWithUTF8String:goodmoring] : @"";
            phrase.ThankYou = (thankyou) ? [NSString stringWithUTF8String:thankyou] : @"";
            phrase.GoodBy = (goodby) ? [NSString stringWithUTF8String:goodby] : @"";
            [phrases addObject:phrase];
        }
        sqlite3_finalize(statement);
    }else {
        NSLog(@"Problem with the database:");
        NSLog(@"%d", sqlResult);
    }
    return phrases;
}

```

4. UIKit を直接利用するスキルの獲得

iPhone アプリのアイデアを実現しようとすると、GUI ツールである StoryBoard を用いただけでは、表現できない場合に遭遇する。そのような場合には、UIKit をプログラムで操作してより複雑な動作をさせることが可能になる。図3にスロットマシン・アプリを紹介する。ちょっとしたアイデアで回り続けるような動作をするアプリを作ることができた。

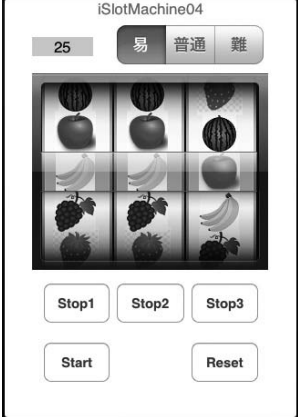
画面例	説明
	<ul style="list-style-type: none"> • このアプリは UIKit の UIPickerView を直接使用する例である。 • 回転速度を0.5秒毎, 0.25秒毎, 0.1秒毎の3つの難易度に設定することができる。 • 4種類のアイテムに5段階の得点を設定し, 得点の合計に3つ揃うと10倍, 2つ揃うと5倍の倍率を掛けて表示するようにした。

図3 スロットマシン・アプリの例

以下のリストは、PickerView を3分割して各列に登録するための画像の配列データを作成している部分である。

```

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    [self.view addSubview:myPickerView];
    UIImage* imageApple=[UIImage imageNamed:@"apple.png"];
    UIImage* imageBanana=[UIImage imageNamed:@"banana.png"];
    UIImage* imageGrapes=[UIImage imageNamed:@"grapes.png"];
    UIImage* imageStrawberry=[UIImage imageNamed:@"Strawberry.png"];
    UIImage* imageWaterMelon=[UIImage imageNamed:@"waterMelon.png"];
    NSArray* components0 = [NSArray arrayWithObjects:imageApple,imageBanana, imageGrapes,
    imageStrawberry,imageWaterMelon, nil ];
    NSArray* components1 = [NSArray arrayWithObjects:imageApple,imageBanana, imageGrapes,
    imageStrawberry,imageWaterMelon, nil ];
    NSArray* components2 = [NSArray arrayWithObjects:imageApple,imageBanana, imageGrapes,
    imageStrawberry,imageWaterMelon, nil ];
    self.dataSource_ = [[NSArray alloc] initWithObjects:components0,components1,components2, nil];
    c1 = 0;
    c2 = 0;
    c3 = 0;
    stopFlag1 = FALSE;
    stopFlag2 = FALSE;
    stopFlag3 = FALSE;
}

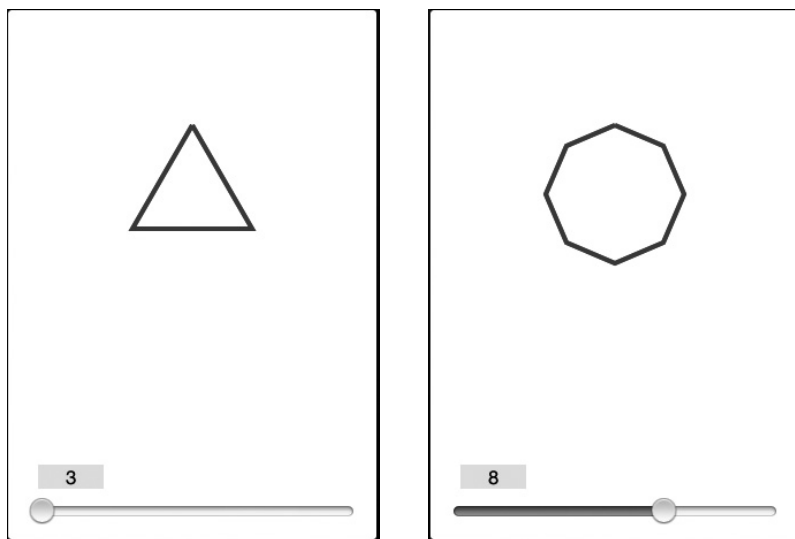
```

5. 魅力的なアプリ開発にはグラフィックス/アニメーションが必須

魅力的なアプリ開発にはグラフィックス/アニメーションに関するプログラミングスキルが必須である。そのためには、コアグラフィックス（Quartz2D）や画像を操作（移動・

回転など)するためのレイヤーの知識が求められる。ゲームプログラミングには、魅力的なアプリに応用できる各種のテクニックが用いられており、学習する価値がある。ただ、OpenGL ES を用いた 3D グラフィクスは、技術レベルが高くプログラミングが極めて複雑となるので、今回の検討から除外した。

図4にスライダの値に応じて多角形の形が変化する iPolygon の例を示した。



- Slider の値を 3 から12まで変化するように設定する。
- Slider の値を変化させると、現在の値を myLabel に表示し、その値に応じた多角形を表示エリア (canvas) に表示する。

図4 スライダの値に応じて多角形の形が変化する iPolygon の例

以下のリストは、多角形の値Nを設定した後、呼び出され描画を行うメソッドである。360°をNで割り、sin, cos 関数を用いて頂点を求めて頂点間で線を引くという単純な処理である。i が 0 の場合は始点にペンを移動し、それ以降は線を追加している。

```

-(void)drawRect:(CGRect)rect{
    CGContextRef context = UIGraphicsGetCurrentContext();
    CGContextSetLineWidth(context, 4.0);
    CGContextSetStrokeColorWithColor(context, [UIColor blueColor].CGColor);
    int x,y;
    for (int i = 0; i <= N; i++) {
        double z = i * 2.0 * 3.1415 / N;
        x = 160 + round(60 * sin(z));
        y = 160 - round(60 * cos(z));
        if (i == 0) {
            CGContextMoveToPoint(context, x,y);
        } else {
            CGContextAddLineToPoint(context, x, y);
        }
    }
    CGContextStrokePath(context);
}

```

6. iPhone アプリ開発中級コースの構成例

表2に iPhone アプリ開発中級コースの目次例を示す。第1章から第2章は、PickerViewを用いたアプリを紹介し、第3章から第4章はデータベースアプリである。第5章から第

表2 iPhone アプリ開発中級コースの目次例

章	目次
第1章	スロットマシン
第2章	機械式計算機シミュレーター（ゼミ生のアイデア開発したもの）
第3章	マルチ言語フレーズ（データベース・アプリ）
第4章	ToDo リスト（データベース・アプリ）
第5章	テンキー電卓
第6章	位置情報アプリ
第7章	マルチタブアプリ
第8章	多角形アプリと描画ツール
第9章	神経衰弱ゲーム(1)
第10章	神経衰弱ゲーム(2)
第11章	落ち物パズル(1)
第12章	落ち物パズル(2)
第13章	脱出ゲーム(1)
第14章	脱出ゲーム(2)
第15章	ゲームグラフィクスとゲームサウンドの作り方

（注）第9章から第15章は、市販テキスト「沼田哲史（2011）『15歳からはじめる iPhone わくわくゲームプログラミング教室』ラトルズ」を使用する予定である。

8章は入門コースに収まらなかったものである。第9章から第15章は、市販テキストを使用する。このテキストの全てのプログラムは、再入力・修正して最新の開発環境でも動作するように確認を行っている。

コース全体に亘って、極力 GUI ツールを使わずにプログラムでアプリを構築するようにした。当面、ゼミ生の希望者に参加してもらって内容の改善をおこなっていききたい。

7. お わ り に

本稿では4年間にわたり少人数教育および IT ビジネス特修実習で改善してきた iPhone アプリ開発入門教育を発展させた中級コースの内容について述べた。中級の iPhone アプリ開発は Apple 社が提供するテンプレートおよびフレームワークに影響されにくい方法で教育し、本質的かつ自由度の高い創造的なアプリを開発できる手段を提供すべきである。

従って、データベースを直接操作する方法、GUI ツールを使わずに UIKit を活用する方法、グラフィクスやアニメーションの利用方法などを網羅的に提供する予定である。

今後、MIPS の更改が 1～2 年後に始まる予定だが、ひとつの教室は38号館のように iMac を30台ほど導入し、Windows と Mac OS X のデュアルブートができるようにすべきである。2009年末頃には、iPhone アプリの数は約10万本程度であったが、2013年6月で90万本となっている。実学教育としての IT ビジネス特修実習コースのひとつとして「iPhone アプリ開発入門(実施中)」、「iPhone アプリ開発中級(本年後期からゼミ生を対象に試行開始)」を充実・改善し、近い将来にすべての学生に公開して学生の IT スキル向上に寄与したい。

参 考 文 献

- Apple (2010), *Core Animation プログラミングガイド*
- Apple (2011), *iOS OpenGL ES プログラミングガイド*
- Apple (2012), *Quartz 2D プログラミングガイド*
- 萩原剛志 (2008) 『Objective-C 2.0』ソフトバンククリエイティブ
- 木下誠 (2009) 『iPhone SDK3 プログラミング大全』アスキーメディアワークス
- 所友太 (2010) 『iPhone プログラミング UIKit 詳解リファレンス』リックテレコム
- 沼田哲史 (2010) 『実践 iPad/iPhone ゲームプログラミング』秀和システム
- 沼田哲史 (2011) 『15歳からはじめる iPhone わくわくゲームプログラミング教室』ラトルズ