下っ端ソフトウェア屋の初めてのAndroidアプリ開発

2010/04/10

日本Androidの会中国支部

松原和也

自己紹介

- **名前：松原和也**
- **所属：とあるIT系メーカー子会社勤務**
- **お仕事：製品の仕様を書いたりプログラムを書いたりしてます**
- Twitter ID：Toro_kun
- Blog：http://switch.dip.jp/toro/blog/
  - URL**が長いので**　http://goo.gl/55Oc
- **使用言語：**C、C++(**主に**MFC)

Androidとの出会い

- 1980年12月　香川県高松市に生まれる
- ~2006年　学生時代に携帯電話でＡＲをする研究をしていた
  - 携帯端末に対する興味はこのころからあった
- 2006年　就職で広島に引越し
- 2009年10月　Androidの存在を知る
  このころから勉強会に参加するようになる
- 2010年1月　アプリを作ったこともないのに無謀にもハッカソンに参加
- 2010年1月末　初アプリ「URL Shortener」を公開

ソフト屋から見たAndroidプラットフォームの魅力

- **オープンソースである**
- **アプリはJavaで書くことができる**
  - **言語的に認知されているから本も充実していて入門しやすい？**
- **端末の各種センサーを制御することができる**
- **実機で動かすことができる**
- **世間的にも注目を浴びてきている**

何かおもしろいことができるかもと希望が持てる

まずアプリ開発初心者になる

- Java SDKとeclipseをインストール
- Android SDKをインストール
- エミュレータでビルトインアプリで遊ぶ
- 1つプロジェクトを作成する
- 素のプロジェクトをエミュレータで動かす
- Javaをある程度学ぶ

初心者から初級者へ

- **参考書を買う**
- Hello, Android**のプログラムをなんでもいいから改造する**
  - **私の場合時計(文字列)表示プログラムを書いた**
- **ある程度、感覚をつかめてきたらハッカソンに参加する**
  - **ハッカソンでチームを組んで**Android**アプリ開発経験者から話を聞く**
- Git**からソースコードをダウンロード**

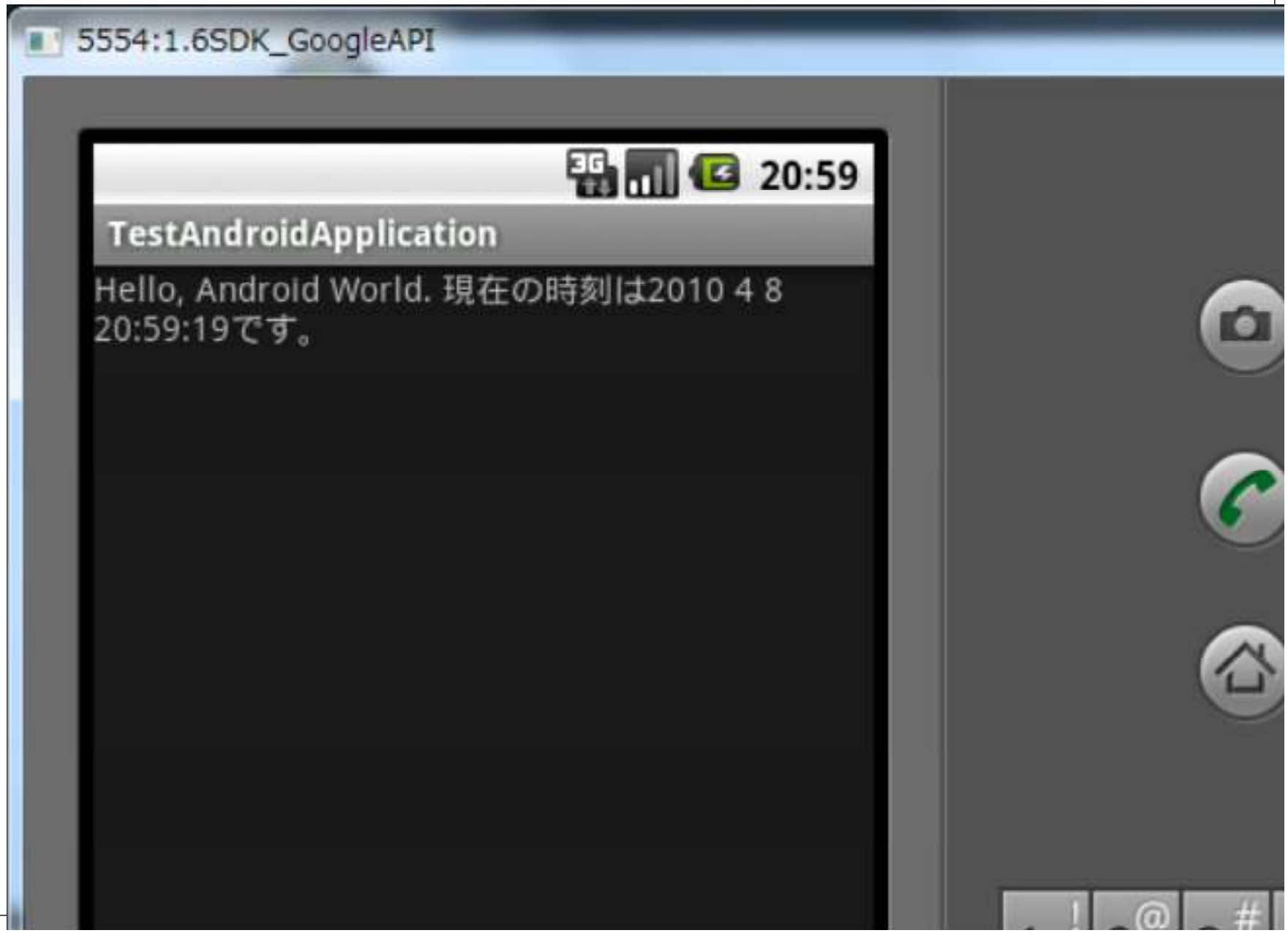
1. 参考書を買う

- 初めて触る技術プラットフォームなのでできるだけ広く深く解説している本を探す
  - WebのGoogle先生を参考にしてもいいが情報がまとまっていない、分散しすぎて分かりづらい
  - 参考書は一通りまとまっているので調べやすい
- 読みやすいところからさらっと読む
- 熟読はしない
  - 熟読は必要になったときに読み返せばいい
- おすすめの参考書は「Google Android プログラミング入門」
  著者：江川崇、他

2. Hello, Androidを改造する

```
public class TestAndroidApplication extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mHandler.removeCallbacks(mUpdateTimeTask);
        mHandler.postDelayed(mUpdateTimeTask, 1000); // 1秒後に開始
    }
    public TextView getTextView(){
          return new TextView(this);
    }
    private Handler mHandler = new Handler();
    private Runnable mUpdateTimeTask = new Runnable(){
        public void run(){
            long millis = SystemClock.uptimeMillis();
            TextView tv = TestAndroidApplication.this.getTextView();
            String testStr;
            Date dt = new Date();
            testStr = "Hello, Android World. 現在の時刻は" + dt.toLocaleString() + "です。";
            tv.setText(testStr);
            setContentView(tv);
            mHandler.postAtTime(this, millis + 1000);
        }
    };
}
```

2. Hello, Androidを改造する



3. ハッカソンに参加する

- **ハッカソンでAndroidアプリ開発経験者がいるのでいろいろ話を聞いて技術を盗む**
  - **そのときに全て理解できなくてもいい**
- **チームを組む**
  - **1人でハッカソンをするよりも楽しい**
- **ハッカソンで作ったソースは共有してもらう**
  - **知らない部分を学べる**
- **アイデアソンに参加するだけでも楽しい**
  - **アイデアを考える訓練にもなる**

4. Gitからソースをダウンロード

- Linux**環境が必要。**
  - Ubuntu**など**Virtual PC**や**VMware**などを使えば追加投資はいらない。**
- Android**の標準アプリのソースも見ることができる**
  - **電話アプリや連絡先アプリ、カレンダーアプリすらも見ることができる**
- Android**アプリ作成の作法を見ることができる**
- **標準アプリで実装されている機能をまねできる**
  - **非公開**API**の「かかってきた電話に出る」コードも見れます**

もう1つのgoogle**先生**

ソースコードをダウンロードするには(参考)

- **以下のパッケージをLinuxにインストール**
  - Git 1.5.4
  - Curl
- **以下のコマンドを実行**
  1. $ mkdir ~/bin
  2. $ export PATH=$PATH:~bin
  3. $ curl http://android.git.kernel.org/repo >~/bin/repo
  4. $ chmod a+x ~/bin/repo
  5. $ mkdir mydroid
  6. $ cd mydroid
  7. $ repo init -u git://android.git.kernel.org/platform/manifest.git
  8. $ repo sync

とはいっても・・・

- いきなりアプリのコードが書けるようにはならない
- アプリのネタもすぐには出てこない
- まずは見た目から習得したほうが入りやすい
- やっぱり見た目は重要
- ユーザビリティやダウンロード数にも影響する

UIをある程度知ろう

- **まずはアクティビティのUIを知ろう**
- **アクティビティのテーマ(背景)**
  - **黒い(標準)** Theme_Black (Theme)
  - **明るい** Theme_Light
  - **透明** Theme_Translucent
  - **ダイアログ風** Theme_Dialog
- **個人的には黒いテーマ(Theme_Black)より明るいテーマ(Theme_Light)のほうが見た目がいいと思う**
- **“フルスクリーン”や”タイトルバーなし”も組み合わせることもできる**

テーマのいろいろ(1)

Theme_Black (Theme)

Theme_Light

テーマのいろいろ(2)

Theme_Dialog

Theme_Translucent

テーマの指定方法

- XMLによる指定(AndroidManifest.xml内)

```
<application android:icon="@drawable/icon"
             android:label="@string/app_name"
             android:theme="@android:style/Theme.Light">    ※ココ
```

<application>タグ、または<activity>タグ内で指定する

- ソースコードによる指定

```
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setTheme(android.R.style.Theme_Light);        ※ココ
    setContentView(R.layout.main);
}
```

setTheme()でテーマを指定します。

アクティビティのレイアウト

- **レイアウトは簡単に言うとウィジェット(ボタンやテキストボックスなど)の配置を定義したものである。**
- **レイアウトファイルは**/res/layout**に配置することで、コードからはリソースインデックス**(R.layout.<**ファイル名**>)**としてアクセス可能である。**
- **ツール系のアプリの場合、静的な画面が多いのでレイアウトファイルで画面を構成したほうが作りやすい**

アプリネタがないかと探していたら

- Twitterでよく使われている短縮URLにgoogleが手を出したというニュースを聞く(2009年12月)
- 「goo.gl」というドメインでやってるとか。
- goo.glを使う短縮URL APIは一応非公開となっている。
  - Google Tool BarやGoogle Chromeのエクステンションから使える。
- しかし、Google Tool Barのスクリプトに JavaScriptでベタに書かれてあったので有志がperlなどに移植している。
  →よし、Androidに移植しよう。

URL短縮サービスって？

- どんなURLもhttp://<ドメイン名>/<数文字の文字列>で短縮してしまうサービス
- なんのため？
  - Twitterなどの文字数の限られた入力項目に本文をできるだけ圧迫しないでURL(リンク)を貼り付けることができる
- どんなサービスがある？
  - bit.ly
  - TinyURL　などが有名

goo.glの短縮URLの作り方(1)

- http://goo.gl/api/urlに以下のデータをPOST
  - user=ユーザ名
  - url=短縮したいURLをURLエンコードした文字列
  - auth_token=認証トークン？

- ユーザ名は「toolbar@google.com」でよい
- 認証トークンはGoogle Tool BarのJavaScriptからアルゴリズムを移植して生成

goo.glの短縮URLの作り方(2)

- **例えば「**http://www.android-group.jp/**」を短縮するなら・・・**
  - http://goo.gl/api/url?user=toolbar@google.com&url=http%3A%2F%2Fwww.android-group.jp%2F&auth_token=75560172099

  をPOSTする。
- **サーバからの応答がステータスコード**CREATED(201)**で返ってくる**
- **レスポンスデータはJSON形式で短縮URLが得られる**

  {"short_url":"http://goo.gl/rMhi"}

  この値から「http://goo.gl/rMhi**」を抜き出す**

注意点

- Internetに接続するということは
  - Internet接続のパーミッションをAndroidManifest.xmlに設定する必要がある
    - <uses-permission android:name="android.permission.INTERNET"/>
  - UIスレッドで時間のかかる処理するとエラーじゃないのにエラーと表示される
    - 時間のかかる処理は別スレッドを立てて処理しましょう

アプリとして一応動いたら

- Android Marketに公開の準備をする
- アプリのアイコンに悩む
- デザイナーではないのでガイドラインにあるようなアイコンなんて作れません・・・
- 立体的な画像の作り方を探すと・・・
  - Inkscapeというフリーソフトで作れる
    - ベクター画像を扱える
    - IllustratorとPhotoshopを買わなくてもいい
- Androidのアイコンをただで(Inkscapeで)描く(ためのリンク)
  - http://goo.gl/OXeE

アイコンデザインガイドライン Android 2.0

- http://developer.android.com/intl/ja/guide/practices/ui_guidelines/icon_design.html

アプリ開発Tips(1)

- targetSdkVersionとminSdkVirsion
  - **マニフェストファイルで開発SDKのバージョンと動作可能最低SDKバージョンを指定できる**

    <uses-sdk android:minSdkVersion="3" />
    <uses-sdk android:targetSdkVersion="4" />

    としておくと、Android 1.5**で動き、**Android 1.6**のAPIも使える**

    ただし、1.5**で動かないAPIを使うときは**android.os.Build.VERSION.RELEASE**の値で処理を振り分ける必要がある**

アプリ開発Tips(2-1)

- **アクティビティのアニメーション(1)**
  - res/values/themes.xml**にカスタムテーマを記述できる**
    - **ここにテーマに適用する**WindowAnimasion**設定の記述することができる**

```
<style name="Theme.Activity"
parent="android:style/Theme.Light">
   <item
name="android:windowAnimationStyle">
@style/Animation.Activity</item>
</style>
```

アプリ開発Tips(2-2)

- **アクティビティのアニメーション(2)**
  - res/values/styles.xmlに各アニメーションの設定ファイルを記述する

```
<style name="Animation.Activity"
parent="android:Animation.Activity">
        <item
name="android:activityOpenEnterAnimation">@anim/activit
y_open_enter</item>
        <item
name="android:activityOpenExitAnimation">@anim/activity
_open_exit</item>
        <item
name="android:activityCloseEnterAnimation">@anim/activit
y_close_enter</item>
        <item
name="android:activityCloseExitAnimation">@anim/activity
_close_exit</item>
</style>
```

アプリ開発Tips(2-3)

- **アクティビティのアニメーション**(3)
  - res/anim/<**アニメーションファイル**>**にアニメーション実態を** XML**形式で記述する**

```
<set
xmlns:android="http://schemas.android.com/apk/res/android"
>
        <alpha android:fromAlpha="0.0" android:toAlpha="1.0"
                android:duration="300"
                android:fillAfter="true" android:fillEnabled="true"
                android:startOffset="200" />
        <scale android:fromXScale="0.0" android:toXScale="1.0"
                android:fromYScale="0.0" android:toYScale="1.0"
                android:pivotX="50%" android:pivotY="50%"
                android:duration="300"
                android:fillAfter="true" android:fillEnabled="true"
                android:startOffset="200" />
</set>
```

アプリ開発Tips(2-4)

- **デモ**

- **参考**
  - Throw Life – ActivityのOpenとCloseをアニメーションさせる
    - http://www.adamrocker.com/blog/289/activity_open_close_animation.html
  - URLが長いので・・・　http://goo.gl/rRzr

まとめ

- アプリ開発はまずはUIを
  - アプリを作る上で見た目は大事
- ハッカソンに参加することでStep Up
  - 先輩開発者からいろいろ教わろう
- 2人のGoogle先生
  - オープンソースであることを存分に利用する
- アプリネタは何気ないところから

ご清聴どうもありがとうございました