Android Bazaar and Conference 2012 Spring

Android 2.x/3.x/4.x対応
アプリ開発Tips

2012.3.24

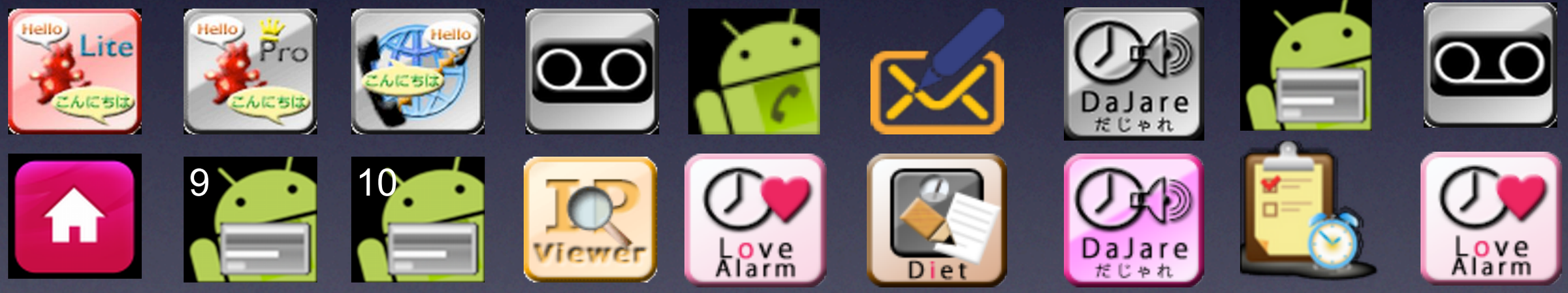
@korodroid（神原　健一）

自己紹介

■氏名：神原　健一（@korodroid）
■活動：iplatform.org(http://www.iplatform.org/)
■所属：NTTソフトウェア株式会社

■主な活動（iplatform.org@プライベート）

■Google Play向けアプリ開発（現在18本）



■Google Developer Day 2011 Tokyo 基調講演デモ
■Android Developer Lab Tokyo 2011 follow-up 5位入賞
■i*deal Competition 2010 ファイナリスト

本日のアジェンダ

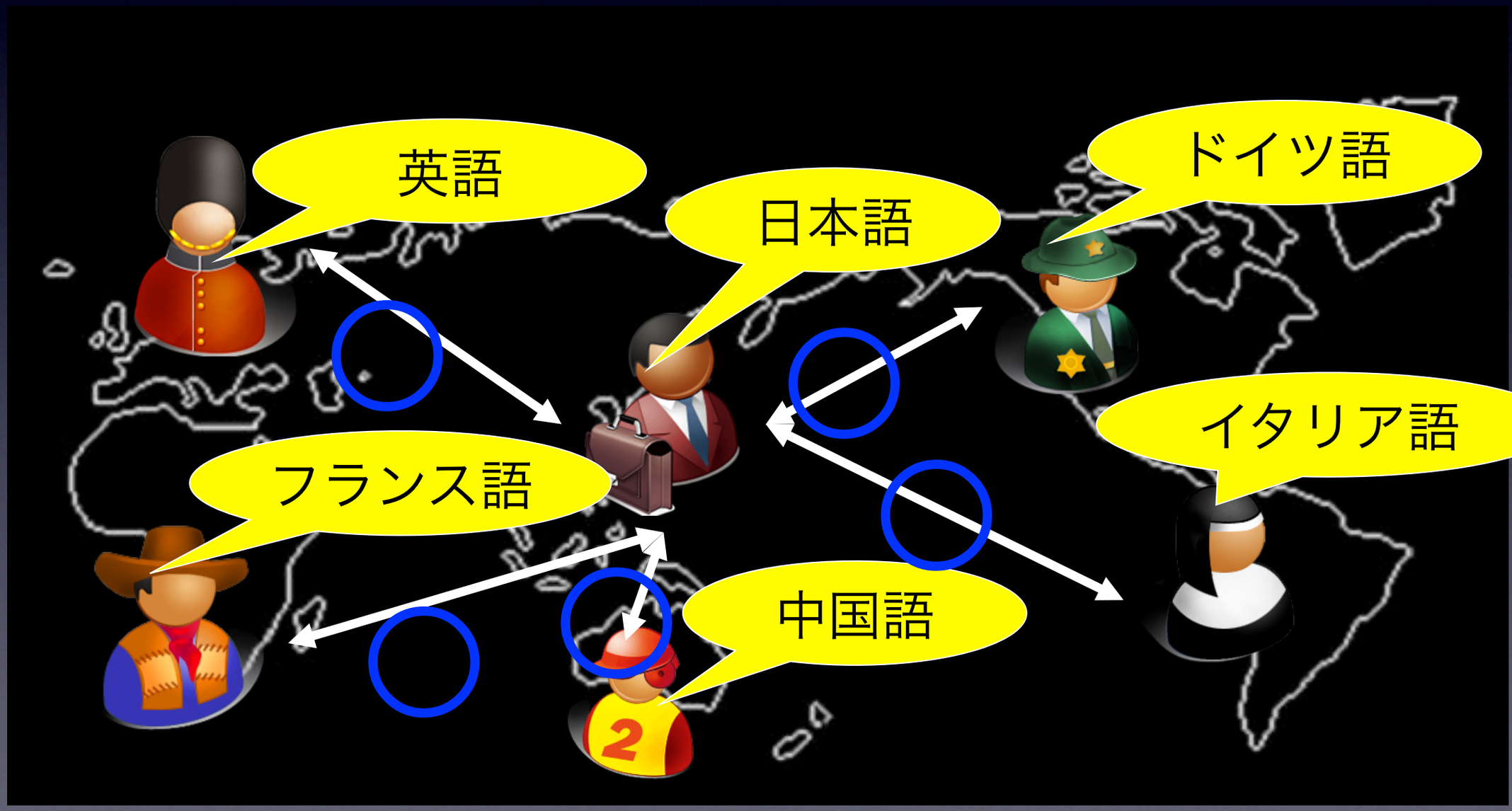
- 本発表の題材とするアプリ
- 背景
- 関連キーワードの復習
- 開発時に特に考慮すべき点

本発表の題材とするアプリ

【セカイフォン】

セカイフォンとは？

6ヶ国語に対応した翻訳電話
（2010/01初版開発以降、20回以上のver.up.）



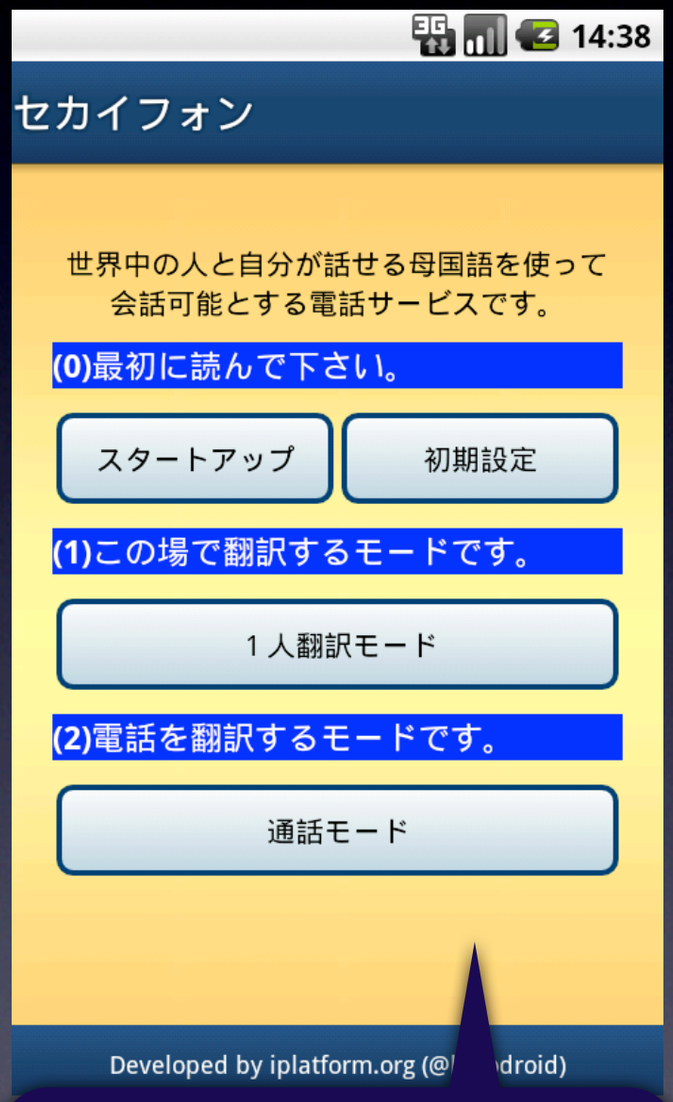
利用シーン①

通話モード【相手の母国語に自動変換】

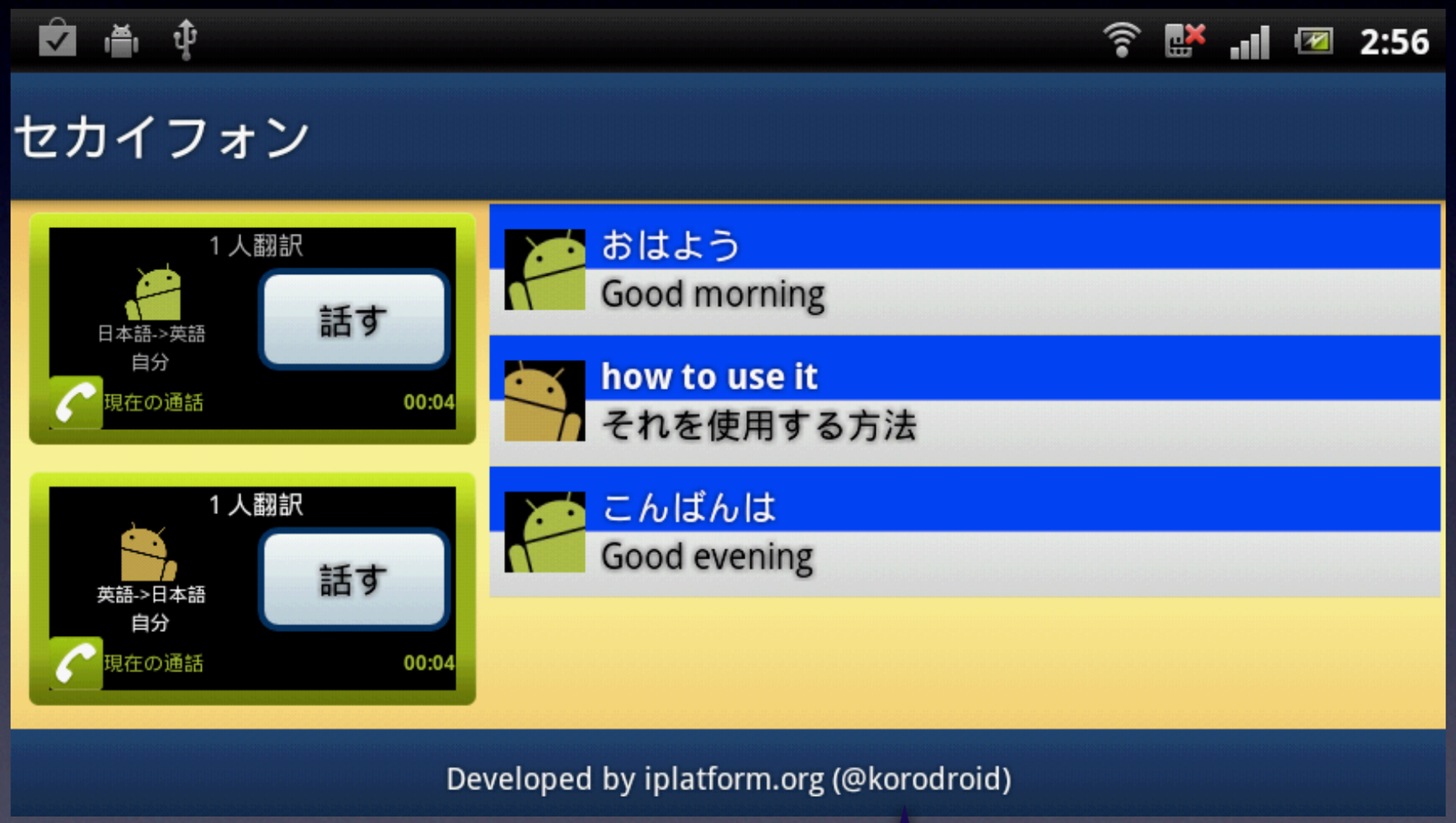
利用シーン②

１人通話モード【会話をその場で変換】

アプリ画面例（Android 2.x）



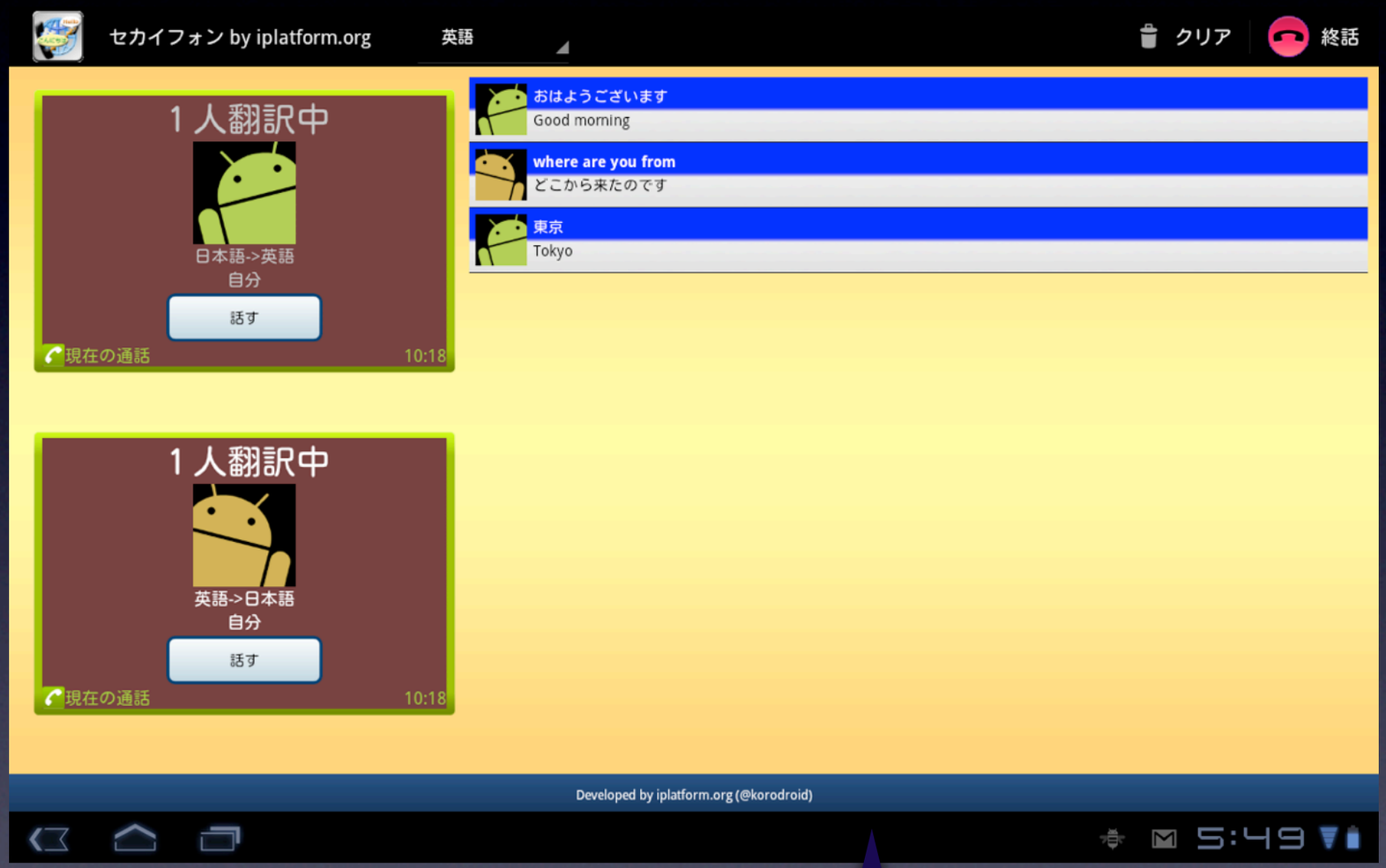
縦画面



横画面

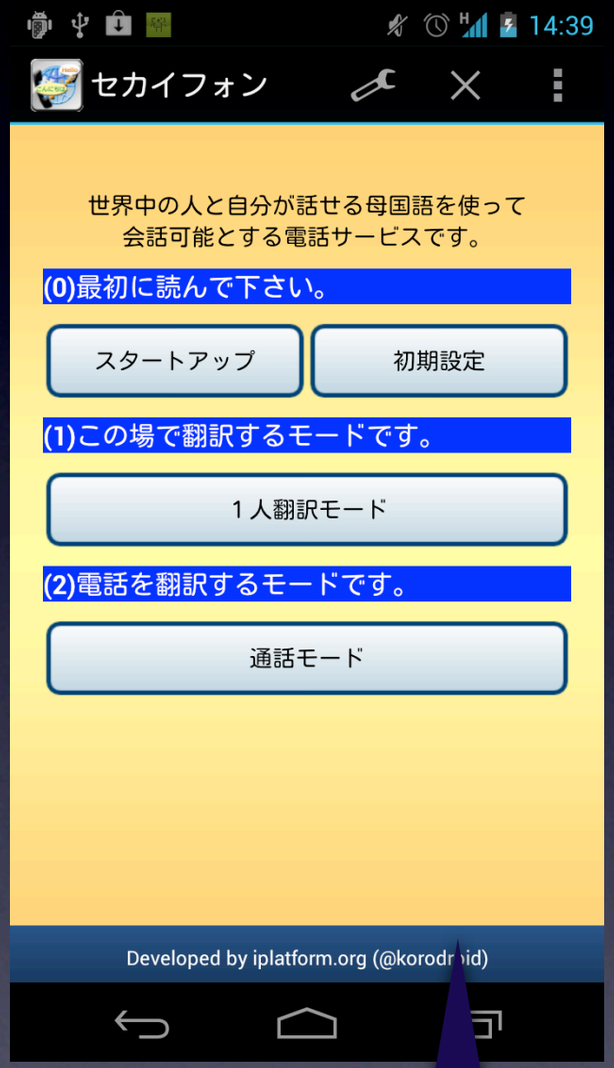
アプリ画面例（Android 3.x）

縦画面



横画面

アプリ画面例（Android 4.x）



縦画面

横画面

背景

背景

アプリ開発のターゲットは？
➡フル対応する場合、現時点では基本的に2.x/3.x/4.x

主にハンドセット

2.x

タブレット

3.x

ハンドセット
＋タブレット

4.x

背景

セカイフォンの2.x/3.x/4.x・マルチデバイス対応時に
気付いた点をお話します（1apkで実現）。

2.x

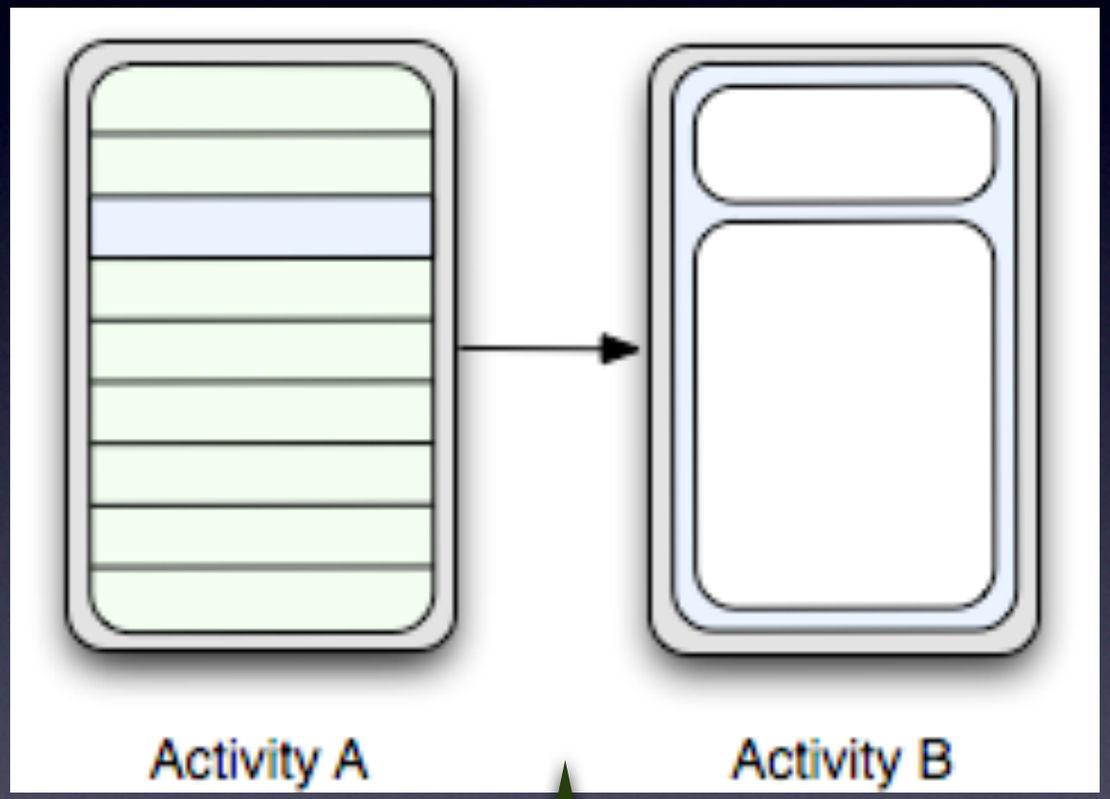
3.x

4.x

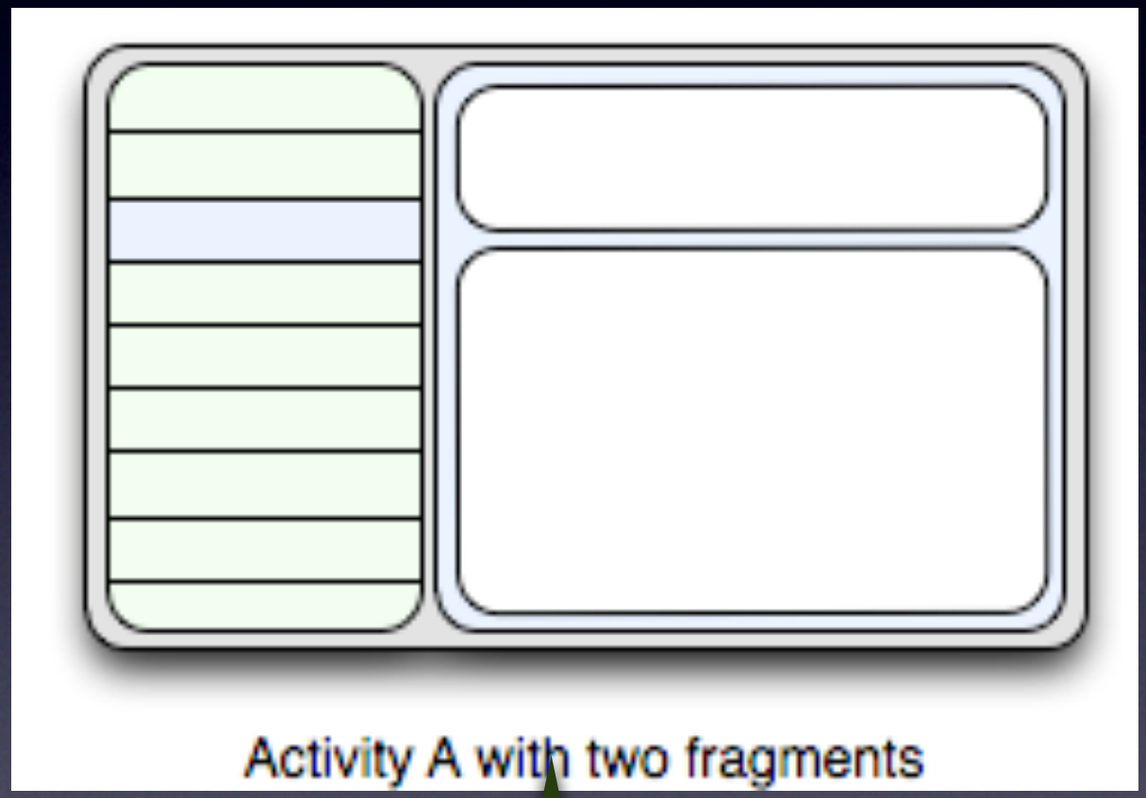
関連キーワードの復習

Fragmentとは？

マルチPaneの画面構成時などに
役立つもの



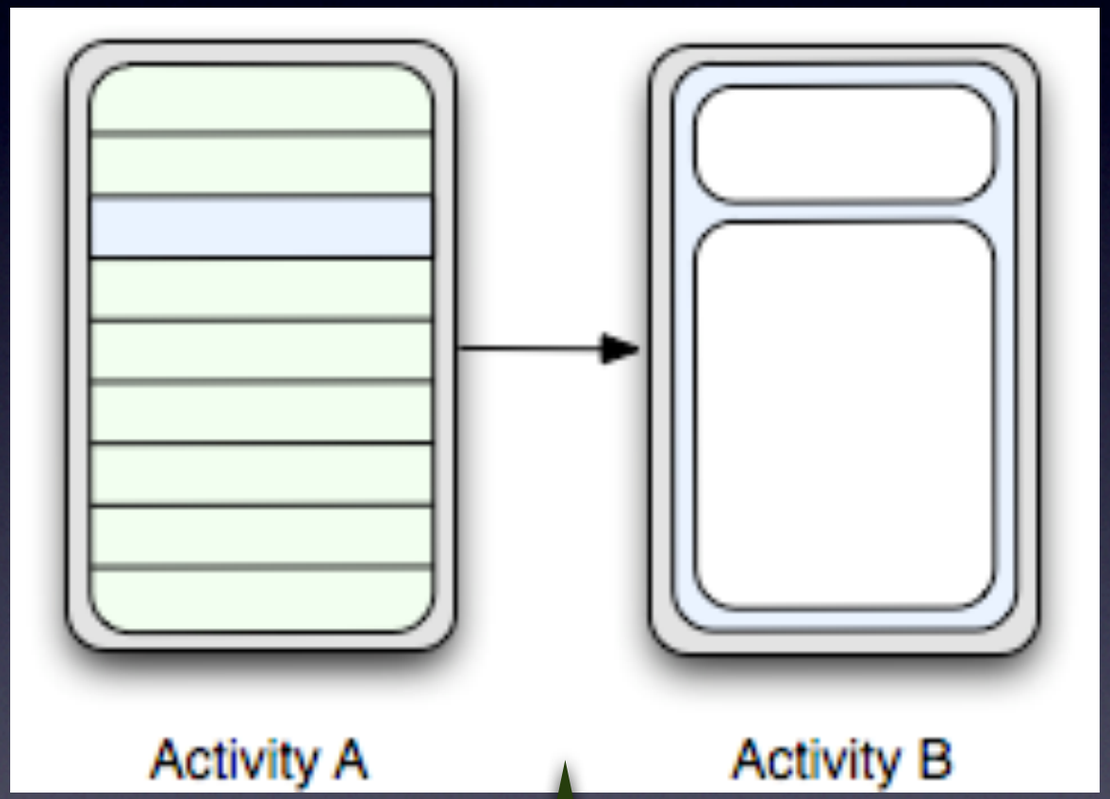
ハンドセットの１例



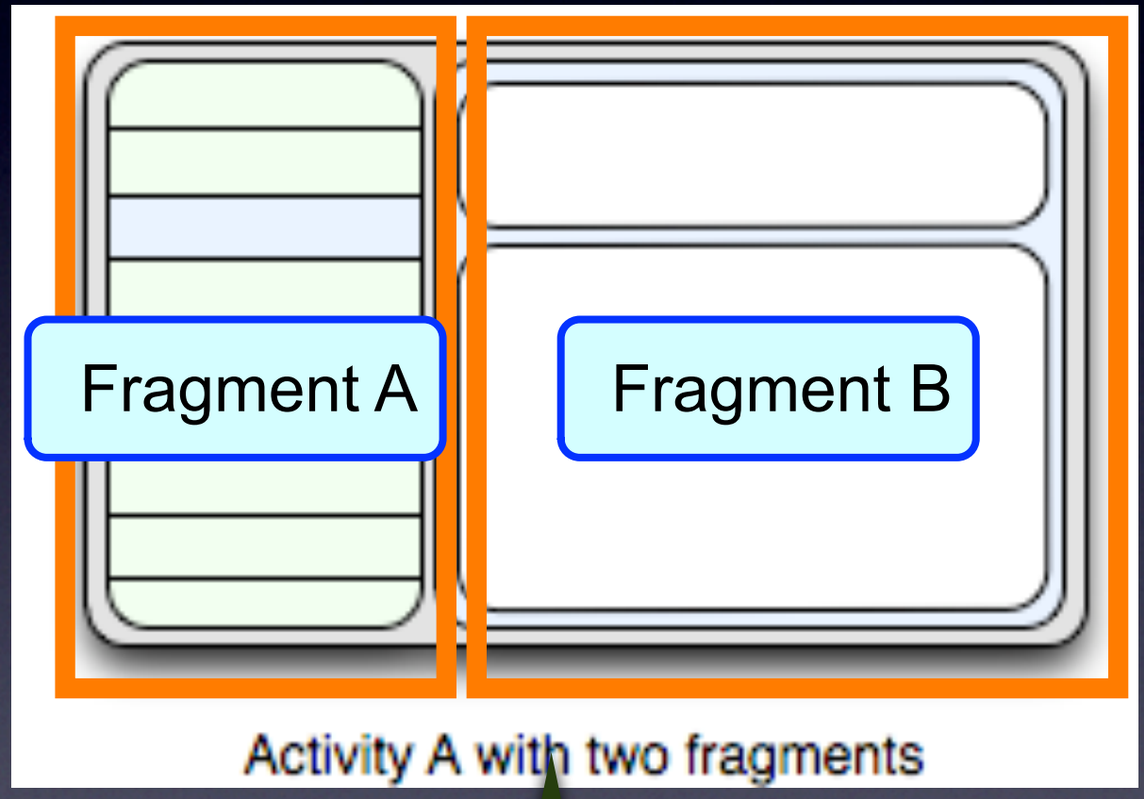
タブレットの１例

Fragmentとは？

マルチPaneの画面構成時などに
役立つもの



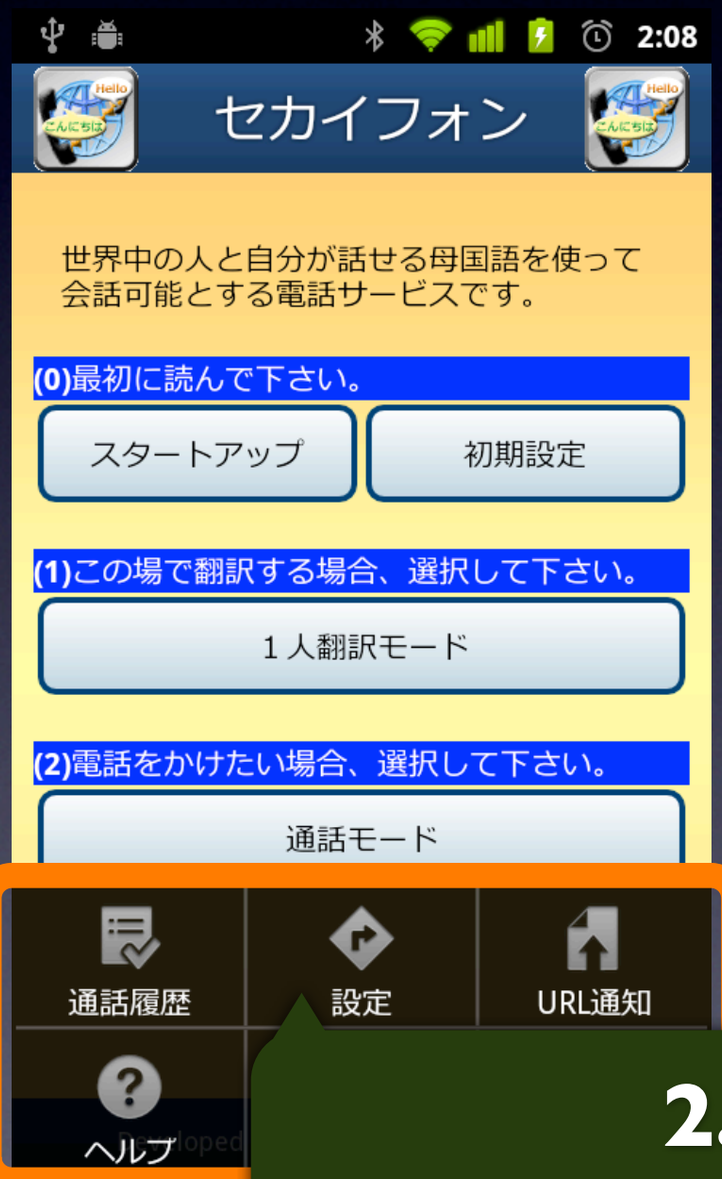
ハンドセットの１例



タブレットの１例

OptionMenuおよびActionBarとは？

「設定」「ヘルプ」など、同画面で
実行したい処理をまとめたもの



2.x
Option Menu

3.x以降
Action Bar

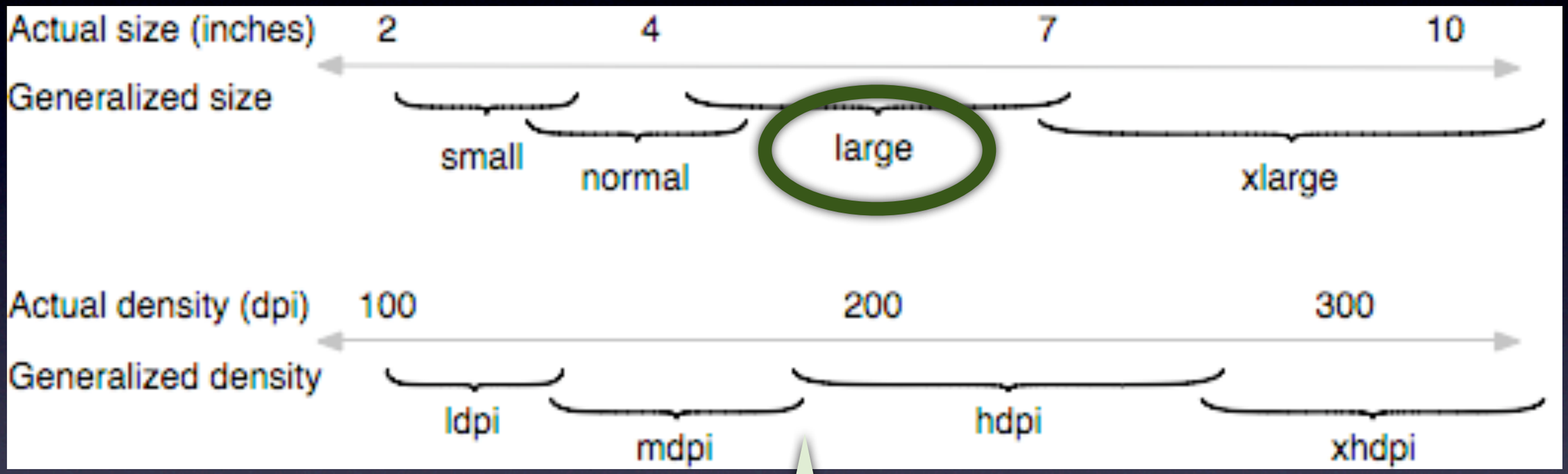
開発時に特に考慮すべき点

1.UI（最適なレイアウト）

2.API（利用可能なAPI）

UI-実現方式①

layout-small/normal/large/xlargeによる分離

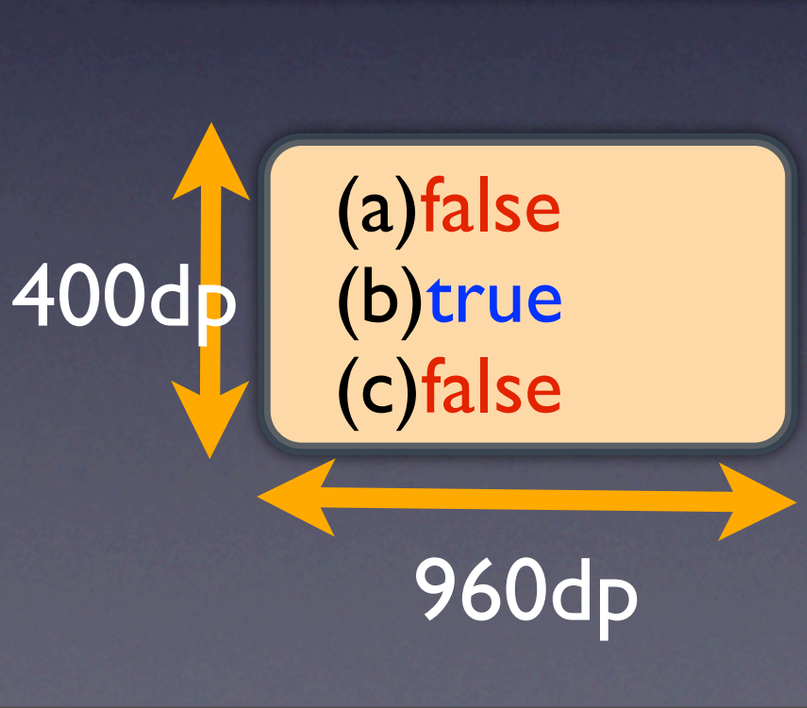


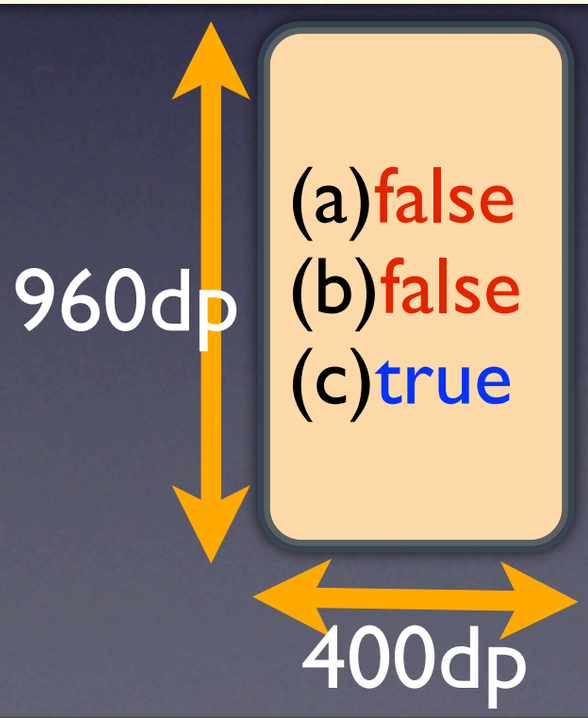
- ○1.6以降であれば、利用可能。
- ×7”タブレットと5”ハンドセットが同じlargeに分類される場合あり。別レイアウトにしたい時に問題に。
- ×3.0以前で適切なグループ分けされない場合あり。

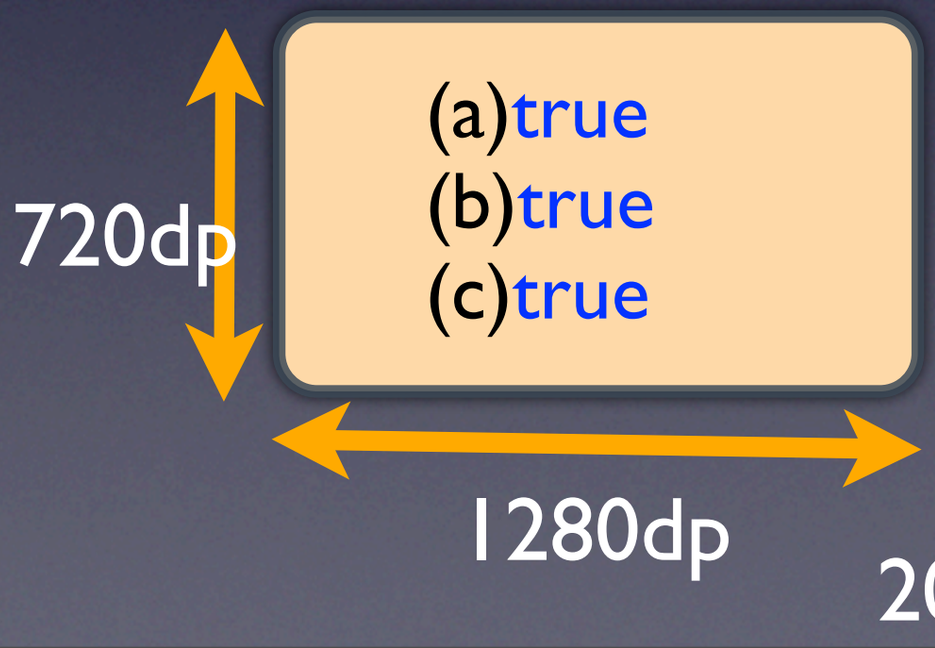
UI-実現方式②

sw<N>dp,w<N>dp,h<N>dpの利用

	概要	補足
(a)layout-sw600dp	縦横ともに600dp以上	画面方向に非依存
(b)layout-w720dp	横が720dp以上	画面方向に依存
(c)layout-h480dp	縦が480dp以上	画面方向に依存





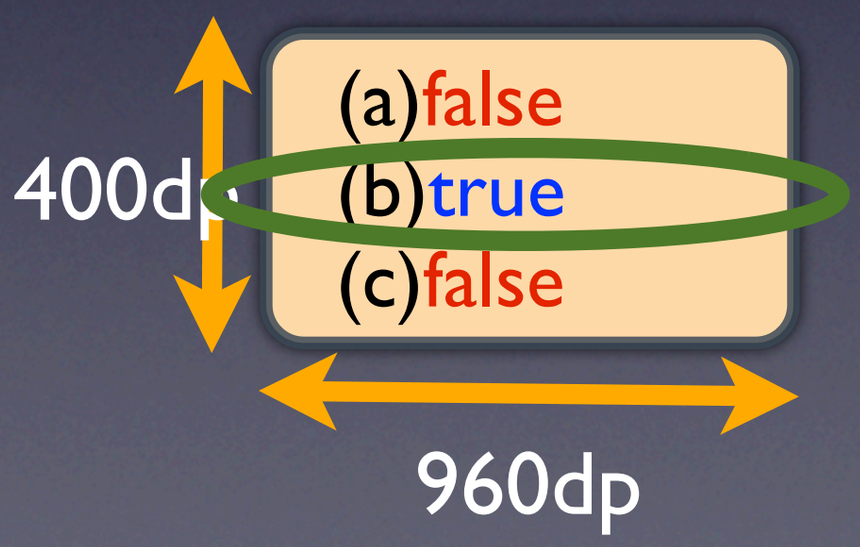


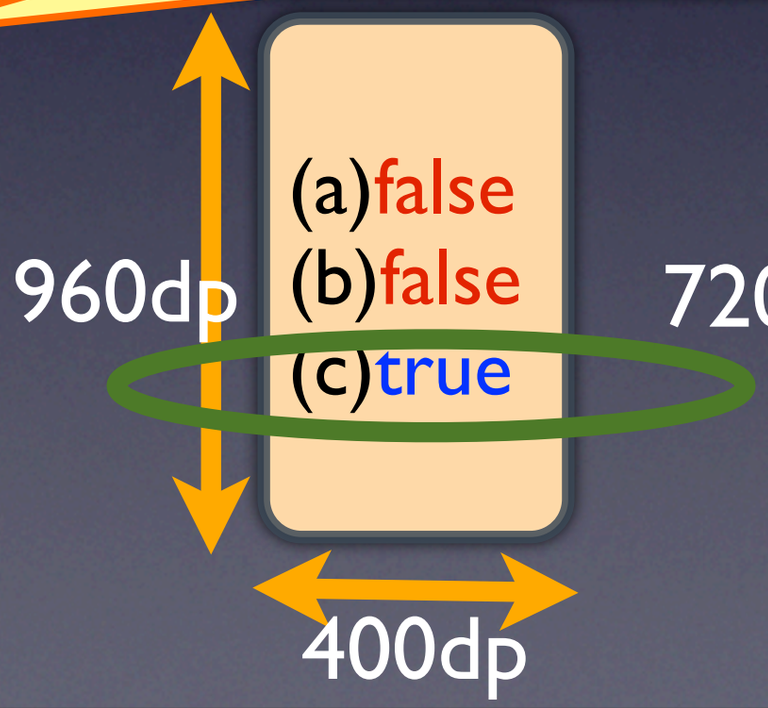
UI-実現方式②

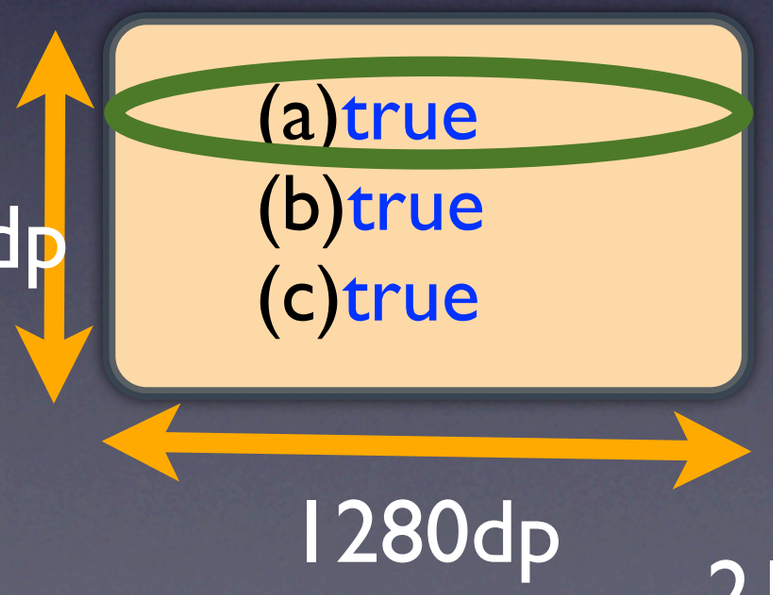
sw<N>dp,w<N>dp,h<N>dpの利用

	概要	補足
(a)layout-sw600dp		
(b)layout-w720dp		
(c)layout-h480dp		

実行する画面特性に応じて、適用されるレイアウトを変更することが可能。







UI-実現方式②

(A)ハンドセットとタブレットの分離例

- res/layout/main.xml
- res/layout-sw600dp/main.xml

(B)ハンドセットとタブレット２種類の分離

- res/layout/main.xml
- res/layout-sw600dp/main.xml
- res/layout-sw720dp/main.xml

(C)タブレット横長とそれ以外の分離例

- res/layout/main.xml
- res/layout-w600dp/main.xml

UI-実現方式②

sw<N>dp,w<N>dp,h<N>dpの利用

	概要	補足
(a)layout-sw600dp	縦横ともに600dp以上	画面方向に非依存
(b)layout-w720dp	横が720dp以上	画面方向に依存
(c)layout-h480dp	縦が480dp以上	画面方向に依存

○Developerサイトで推奨されている。

×3.2以降でしか利用できない。

※画面方向によって、dpが変わるケースがあるので注意（GN：縦長360x592(dp)、横長598x360(dp)）

UI-実現方式③

①・②の併用方式

- layout/main.xml←以下２ヶ以外の場合
- layout-sw600dp/main.xml←3.2以降タブレット
- layout-xlarge/main.xml←主に3.0/3.1タブレット

○1.6以降であれば利用可能。

○前述方式より精度を向上可能。v11等併用もあり。

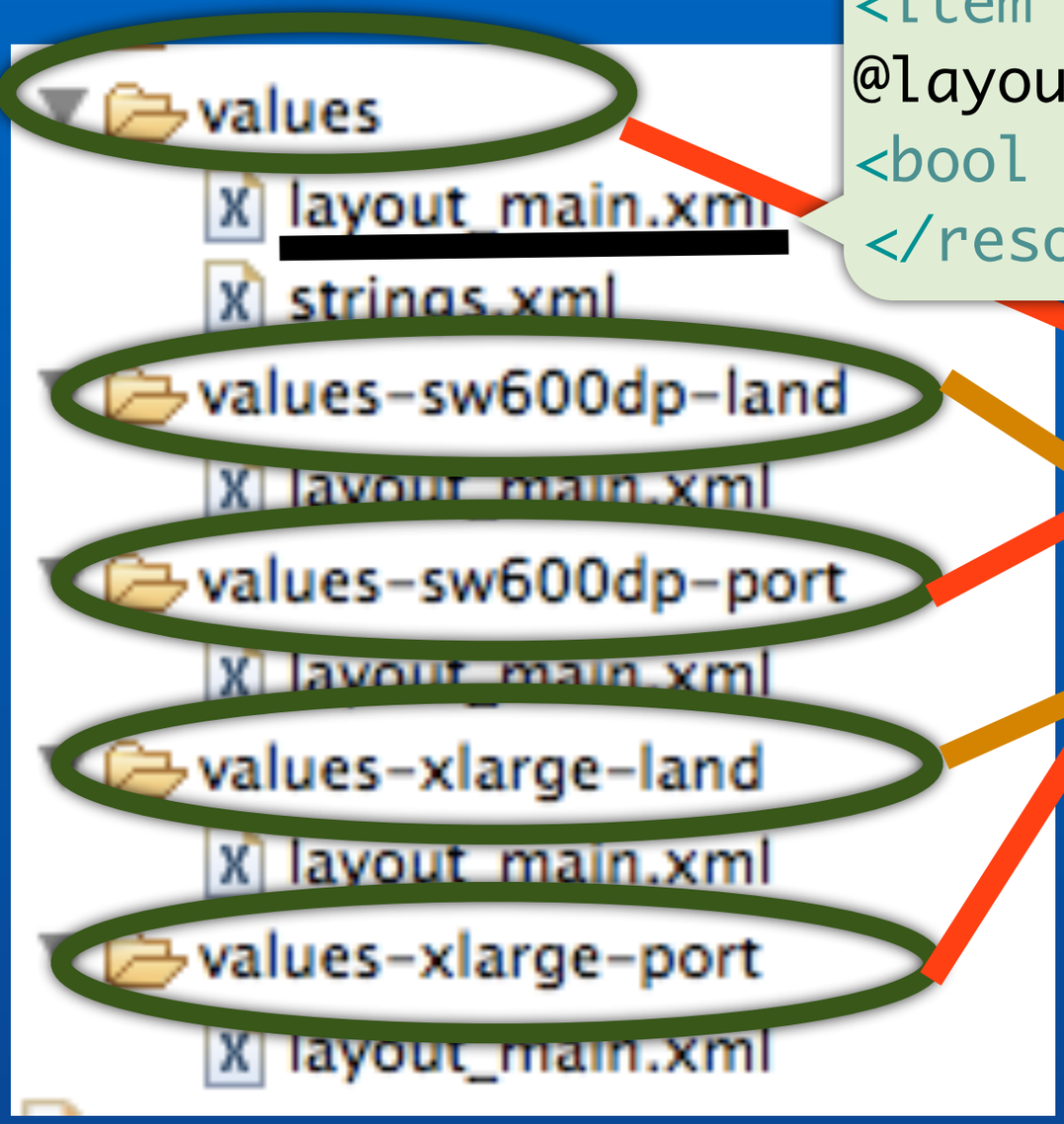
△同じレイアウトの場合も、同じファイルを複数配置。

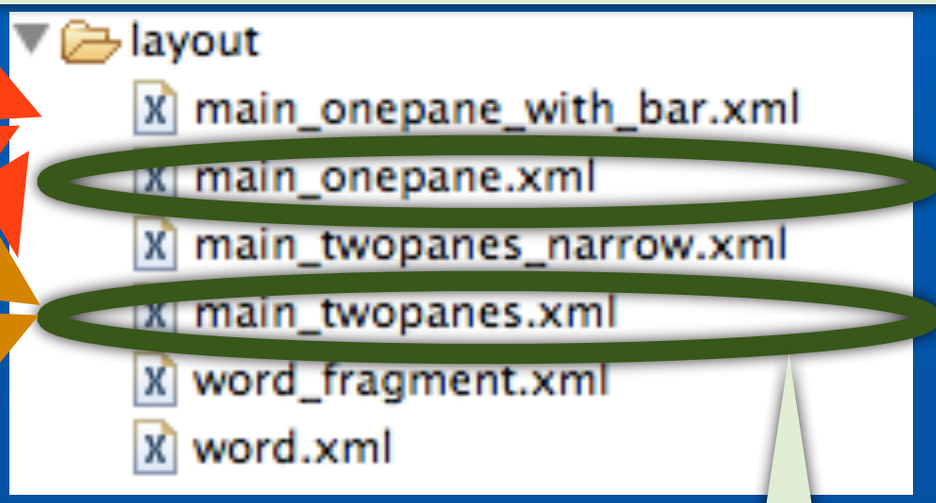
△Javaソース側のロジックも複雑に。

UI-実現方式④

方式③ベースでLayout Aliases活用

```
<resources>
<item name="main_layout" type="layout">
@layout/main_onepane</item>
<bool name="has_two_panes">false</bool>
</resources>
```





- 1pane、2paneなどレイアウトを必要なパターン数のみ準備

UI-実現方式④

方式③ベースでLayout Aliases活用

▪values-xxxフォルダに格納するファイル

```
<resources>
 <item name="main_layout" type="layout">
        @layout/main_onepane</item>
 <bool name="has_two_panes">false</bool>
</resources>
```

▪Java側でのマルチPane判定

```
Resources res = getResources();
boolean hasTwoPanes =
res.getBoolean(R.bool.has_two_panes);
```

判定ロジックはシンプル

API（Fragment）

Support Packageにより2.xでも利用可能

	2.x	3.x	4.x
Support Packageなし	×	○	○
Support Packageあり	○	○	○

- Fragmentベースの実装をしておくことで、再利用性の向上。
- Support Packageの利用有無により実装が変わるので注意。
  - 変更例１）android.app.Fragment->android.support.v4.Fragment
  - 変更例２）Activity->FragmentActivity
  - など幾つかの変更が必要。

API（OptionMenu/ActionBar）

AndroidManifestの内容で振る舞い変化

	2.x	3.x
minSdk=なし＆targetSdk=なし	○	×(NG1)
minSdk=8＆targetSdk=8	○	△(NG2)
minSdk=8＆targetSdk=11	○	○

8:Android2.2, 11:Android3.0

API（OptionMenu/ActionBar）

AndroidManifestの内容で振る舞い変化

	2.x	4.x
minSdk=なし＆targetSdk=なし	○	×(NG3)
minSdk=8＆targetSdk=8	○	△(NG4)
minSdk=8＆targetSdk=11	○	△/○

8:Android2.2, 11:Android3.0

NG3

NG4

ハンドセットでの
ActionBarを
どう評価するか次第

API（OptionMenu/ActionBar）

AndroidManifestの内容で振る舞い変化

	2.x	3.x	4.x
minSdk=なし＆targetSdk=なし	○	×	×
minSdk=8＆targetSdk=8	○	△	△
minSdk=8＆targetSdk=11	○	○	△/○

8:Android2.2, 11:Android3.0

2.x

3.x

4.x

API（OptionMenu/ActionBar）

AndroidManifestの内容で振る舞い変化

	2.x	3.x	4.x
minSdk=なし＆targetSdk=なし	○	×	×
minSdk=8＆targetSdk=8	○	△	△
minSdk=8＆targetSdk=11	○	○	△/○

- JavaソースやXMLが一緒でもManifestで挙動変化
  (上記はあくまで設定の１例（正解はCaseByCase）)。
  ➡オフィシャルBlogに、関連記事あり。
- ICSではSplit ActionBar(画面下部にActionBar表示)も利用可能。採用可否を合わせて検討。

API（OptionMenu/ActionBar）

OptionMenuとActionBarの共存

	2.x	3.x	4.x
OptionMenu	○	○	○
ActionBar	×	○	○

- 2.xでActionBar使えない（Support Packageでも）
  - ➡ActionBarはあきらめてOptionMenuにする
    or 自前で実装する等（ActionBarCompatが参考に）
- 操作性の観点だと、個人的にはActionBarがおすすめ。
- AndroidManifest以外に、Styleによる影響もあり！

その他のTips

1. **wrap_content／fill_parent(match_parent)のみで画面を構成する。**
   ➡具体的なサイズを指定する場合は、dp、sp活用を。

2. **（必要に応じて）画面解像度別の画像を準備する。**
   ➡Android側に制御を任せることも可能だが、拡大縮小に伴い、表示が汚くなる場合がある。

3. **Fragmentをうまく使う。**
   ➡色々な画面でFragmentを再利用できるようにする。
   ➡ActivityとFragmentの結合度縮小。再利用性向上。
   ➡Fragment→Fragment呼出し非推奨。コールバックIF。

その他のTips

4. **ディメンションを活用する。**

➡画面の大きさ別にパーツサイズを切り替えたい場合

- *values/dimens.xml*←以下２ヶ以外の場合
- *values-sw600dp/dimens.xml*←*3.2*以降タブレット
- *values-xlarge/dimens.xml*←主に*3.0/3.1*タブレット

```
<resources>
<dimen name="label_width">160dip</dimen>
<dimen name="label_height">32dip</dimen>
</resources>
```

5. **1apk or マルチapkの方針を判断する。**

➡開発や保守工数に影響があるため、要検討。

参考文献

Android公式Developer向けサイト

- **Supporting Tablets and Handsets**
  http://developer.android.com/guide/practices/tablets-and-handsets.html

- **Supporting Multiple Screens**
  http://developer.android.com/guide/practices/screens_support.html

- **Supporting Different Screen Sizes**
  http://developer.android.com/training/multiscreen/screensizes.html

おわりに

ご清聴ありがとうございました。