

お洒落なAndroidアプリ開発のすゝめ

出典一覧

- **デザイナーに知っていて欲しい10個のAndroidのUIルール – ReDo**

<http://greety.sakura.ne.jp/redo/2011/06/10androidui.html>

- **9patchを覚えよう！ - 明日の鍵**

<http://d.hatena.ne.jp/tomorrowkey/20110501/1304245217>

Androidアプリ開発の基礎知識

Androidアプリ開発について



□ 画面全体: Activity

□ UIコンポーネント

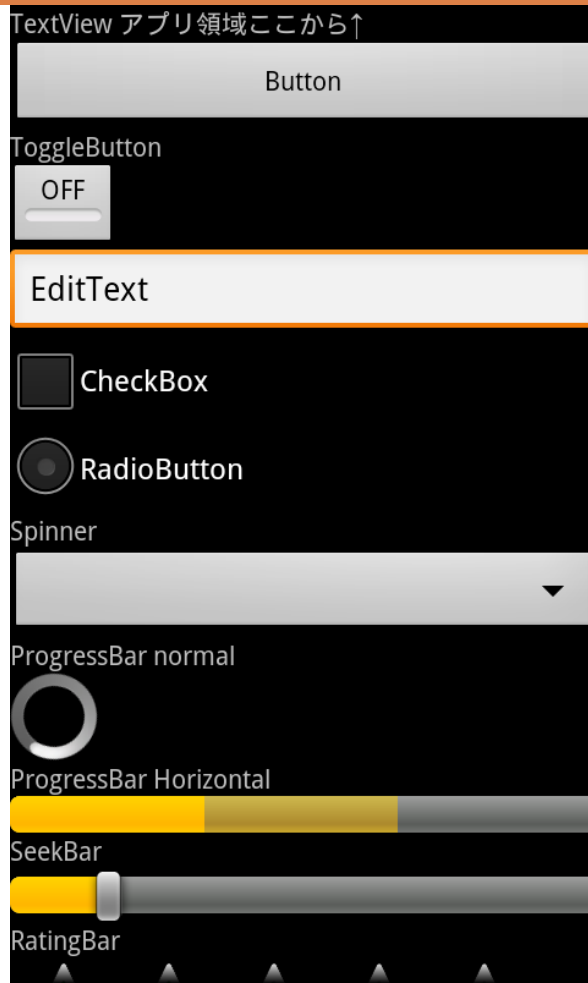
- ・テキストボックス
- ・ラジオボタン
- ・ボタン
- ・タブ
- ・リスト

AndroidのUIデザイン

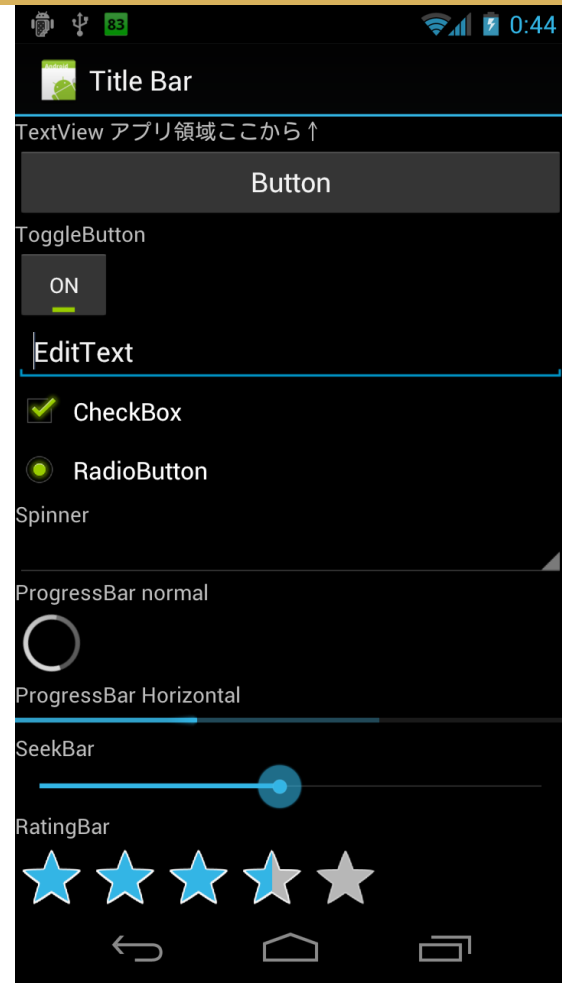
出典: デザイナーに知っていて欲しい10個のAndroidのUIルール - ReDo

Android標準UI

Gingerbread (2.3.x)



Ice Cream Sandwich (4.x)



UIデザインの変更方法

- 個々のUIコンポーネントに変更を施す

この方法では、UIを変更するには全てのUIコンポーネントを書き換える必要がある

保守性を考えると、ピンポイントでデザインを変更したい場合を除いて、この方法は避けるべき

では、どうすべきなのか？

UIデザインの変更方法

- Styleを定義して、UIコンポーネントに適用
- StyleをThemeとしてまとめて、Activityに適用

UIコンポーネント自体とデザインを分けておくことで
変更が容易になる

WebページにおけるHTMLとCSSの関係と同じ？

Styleについて

- UIの見た目を定義するファイル
- XMLで記述
- これをUIパーツやActivityに適用して使う

- Styleを複数まとめたものをThemeと呼ぶ
- Themeという機能があるわけではない

実際にやってみよう



























変更する部分

- タイトルバー
- テキストボックス
- ラジオボタン
- ボタン

これらをICS風に改造

準備

- ▶  PokemonTotalTools
 - ▶  src
 - ▶  gen [Generated Java Files]
 - ▶  Android 2.1
 - ▶  Android Dependencies
 - ▶  assets
 - ▶  bin
 - ▶  res
 - ▶  anim
 - ▶  color
 - ▶  drawable
 - ▶  drawable-hdpi
 - ▶  drawable-ldpi
 - ▶  drawable-mdpi
 - ▶  drawable-xhdpi
 - ▶  layout
 - ▶  values
 - ▶  colors.xml
 - ▶  integers.xml
 - ▶  strings.xml
 - ▶  styles.xml
 - ▶  AndroidManifest.xml
 - ▶  proguard-project.txt
 - ▶  project.properties

□ Project/res/valuesに
style.xmlを作成

準備

□ styles.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources
```

```
xmlns:android="http://schemas.android.com/apk/res  
/android">
```

~ここにStyle(Theme)を記述~

```
</resources>
```

タイトルバー

まずは、適当な名前でもStyleを定義する

```
<style name="WindowTitleBackground">↓  
    <item name="android:background">@drawable/title_bar</item>↓  
</style>↓
```

background属性に背景画像(title_bar)を適用

※drawableはアプリ内の画像リソースを表す

タイトルバー

次に、StyleをまとめるThemeを定義する

```
<style name="Theme">↓  
  <item name="android:windowTitleBackgroundStyle">@style/WindowTitleBackground</item>↓  
</style>↓
```

windowTitleBackgroundStyle属性に先程定義したStyleの
WindowTitleBackgroundを適用する

タイトルバー

最後に、AndroidManifest.xmlでThemeを適用する

Name*

.MainActivity

参照...

Theme

@style/Theme

参照...

Label

@string/app_name

参照...

2番目の項目に先程のThemeを指定する

タイトルバー

BEFORE



AFTER



テキストボックス

Styleを定義

```
<style name="Widget.EditText" parent="@style/Widget">↓
  <item name="android:textAppearance">?android:attr/textAppearanceMediumInverse</item>↓
  <item name="android:textColor">@color/primary_text_light</item>↓
  <item name="android:gravity">center_vertical</item>↓
  <item name="android:background">@drawable/edit_text_holo_dark</item>↓
  <item name="android:focusable">>true</item>↓
  <item name="android:focusableInTouchMode">>true</item>↓
  <item name="android:clickable">>true</item>↓
  <item name="android:maxLength">9000</item>↓
</style>↓
```

Themeに追加

```
<style name="Theme">↓
  <item name="android:windowTitleBackgroundStyle">@style/WindowTitleBackground</item>↓
  <item name="android:editTextStyle">@style/Widget.EditText</item>↓
</style>↓
```

テキストボックス

BEFORE



AFTER



ボタン類へのStyle適用

BEFORE



AFTER



ダイアログへのStyle適用

BEFORE

PokemonTotalTools

名前、わざ名、特性

No.9 : かみなりパンチ

タイプ	でんき
分類	物理
対象	通常
威力	75
命中	100%
PP	15

10%の確率で相手を『まひ』状態にする。とくせい『てつのごぶし』の時、威力が1.2倍になる。

OK

8	れいとうパンチ	こおり	物理	75	100
9	かみなりパンチ	でんき	物理	75	100

AFTER

PokemonTotalTools

名前、わざ名、特性

No.9 : かみなりパンチ

タイプ	でんき
分類	物理
対象	通常
威力	75
命中	100%
PP	15

10%の確率で相手を『まひ』状態にする。とくせい『てつのごぶし』の時、威力が1.2倍になる。

OK

8	れいとうパンチ	こおり	物理	75	100
9	かみなりパンチ	でんき	物理	75	100

9patch画像

出典:9patchを覚えよう！ - 明日の鍵

9patch画像とは

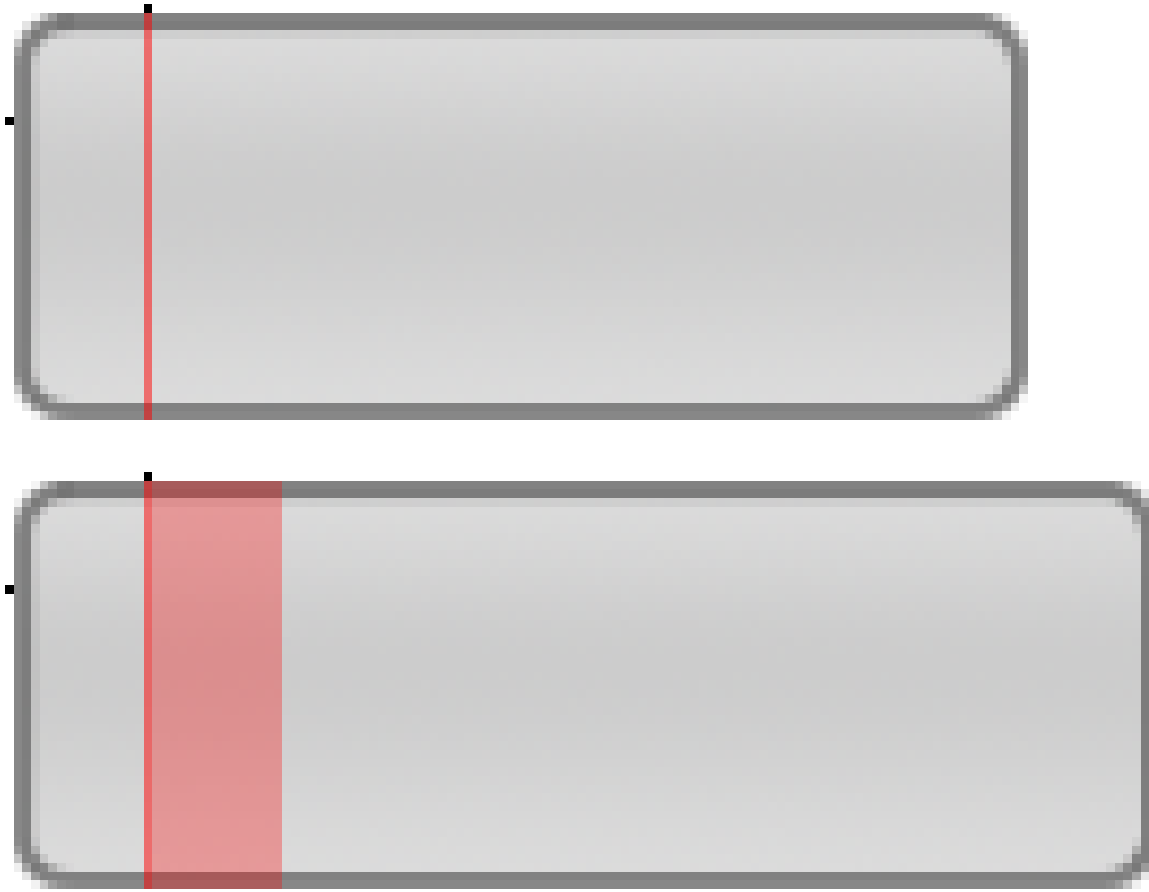
- 容量制限のある携帯端末での画像容量削減法
- 小さな画像を高品位に拡大できる
- 命名規則: image.9.png
- 一番外側のピクセルで拡大する範囲を指定する

言葉で説明しても分かりづらいので



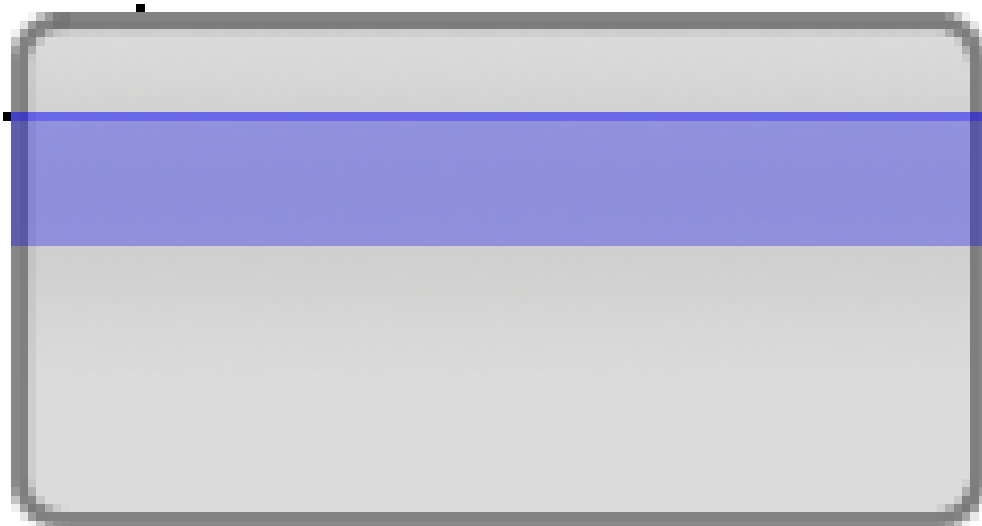
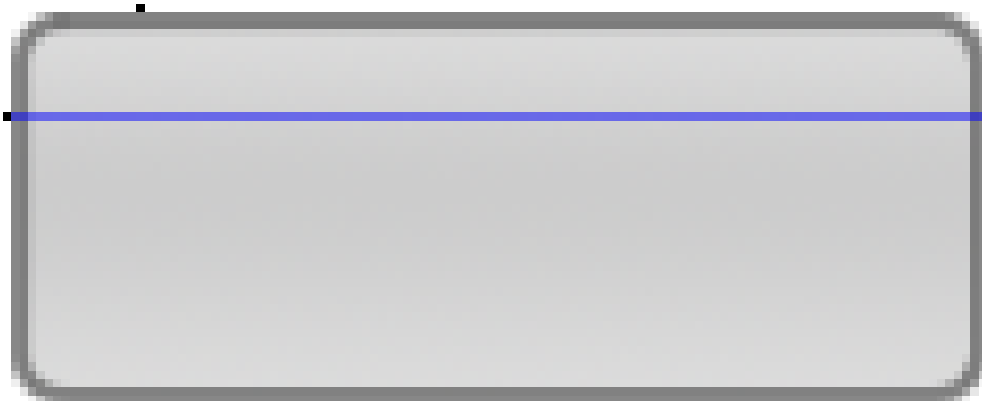
左上に黒い点があるのが、分かりますでしょうか？

9patch画像



赤線のピクセルが拡大時にコピーされる

9patch画像



青線のピクセルが拡大時にコピーされる

Androidにおける9patch画像

- background属性等に9patch画像を指定することで自動的に拡大を行ってくれる
- 先程の例では、ラジオボタン以外で9patch画像を使用しています

まとめ

まとめ

- AndroidのUIデザインではStyleを！
- Themeも活用しよう
- 9patch画像でのアプリケーション容量の削減

以上で終わりです

ご清聴ありがとうございました

これからAndroidアプリケーションの開発をしようと思っ
ている方、すでにやっている方の参考になれば
嬉しいです