

日本Androidの会 浜松支部 第4回ミーティング Aug 13, 2011

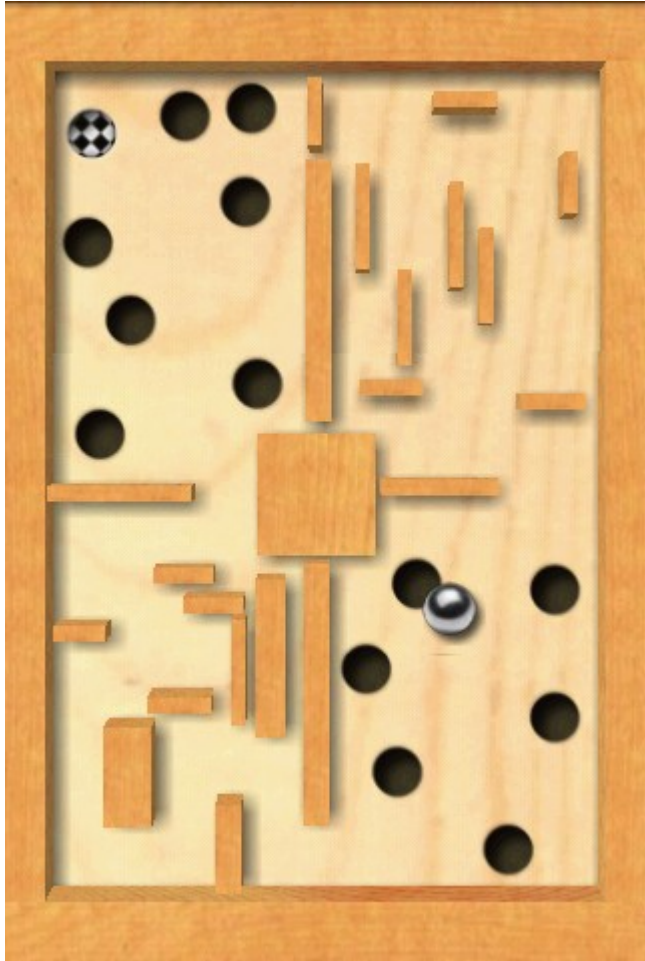
UIカスタム入門

有山圭二（日本Androidの会 関西支部）

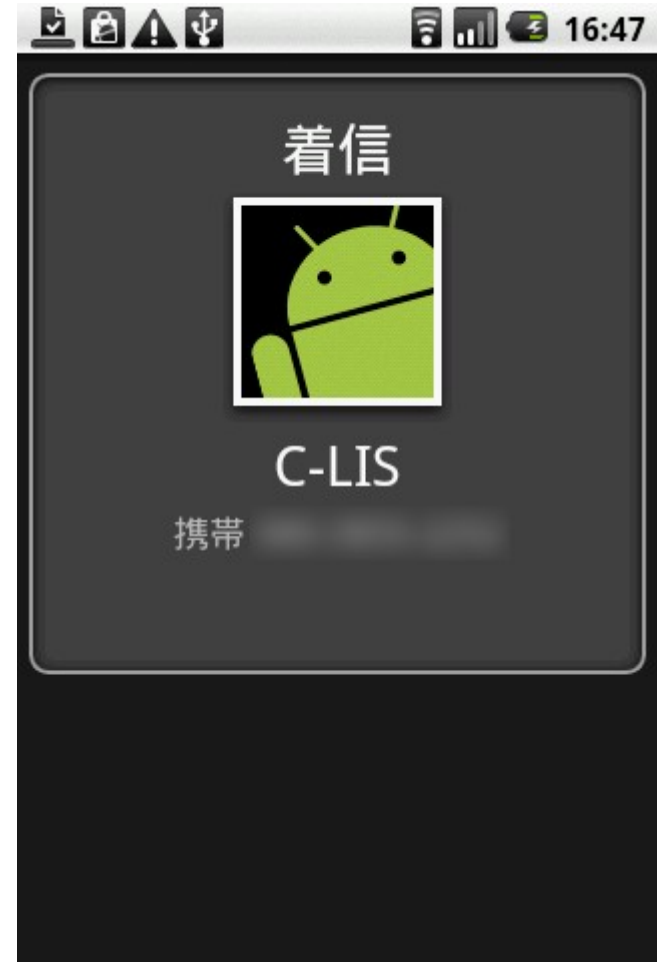
Interlude

アクティビティのライフサイクル

■ ゲーム中に電話がかかってきたの巻



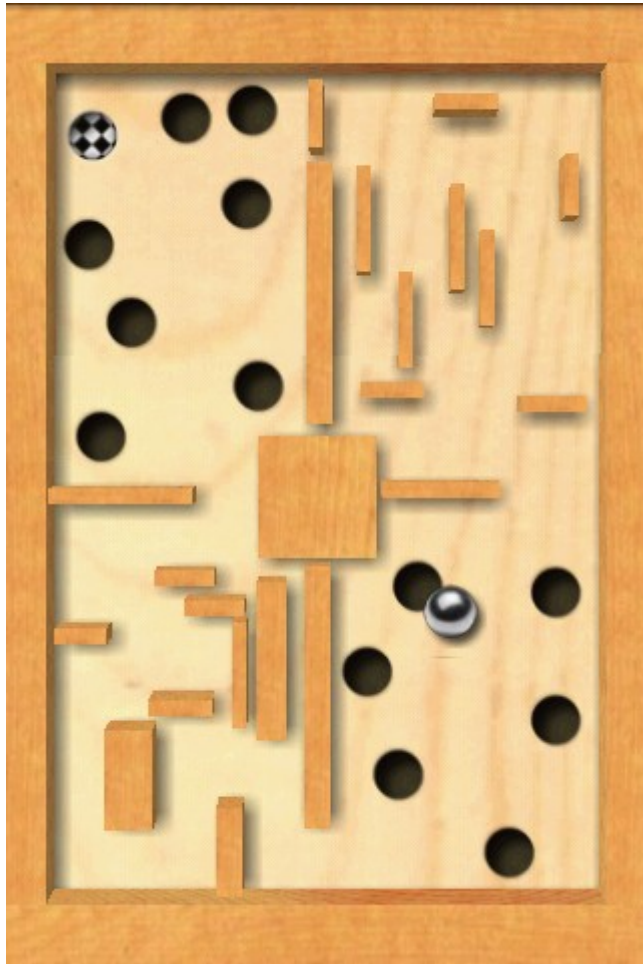
Labyrinth Lite



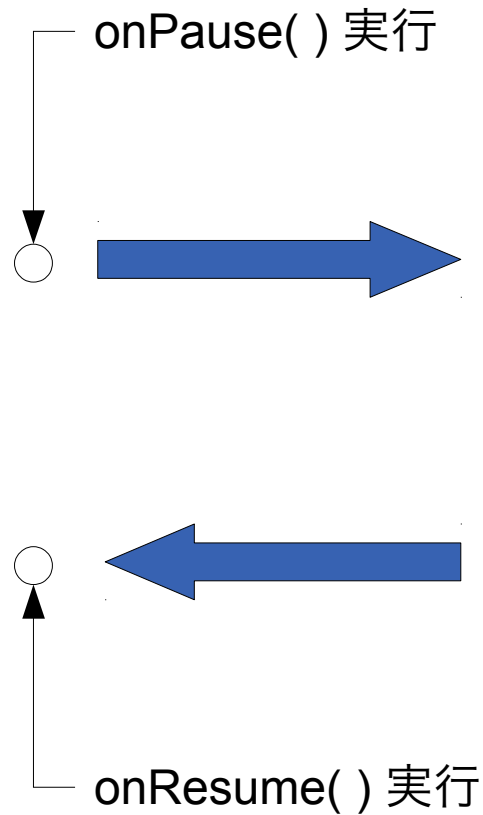
■ アクティビティのライフサイクル

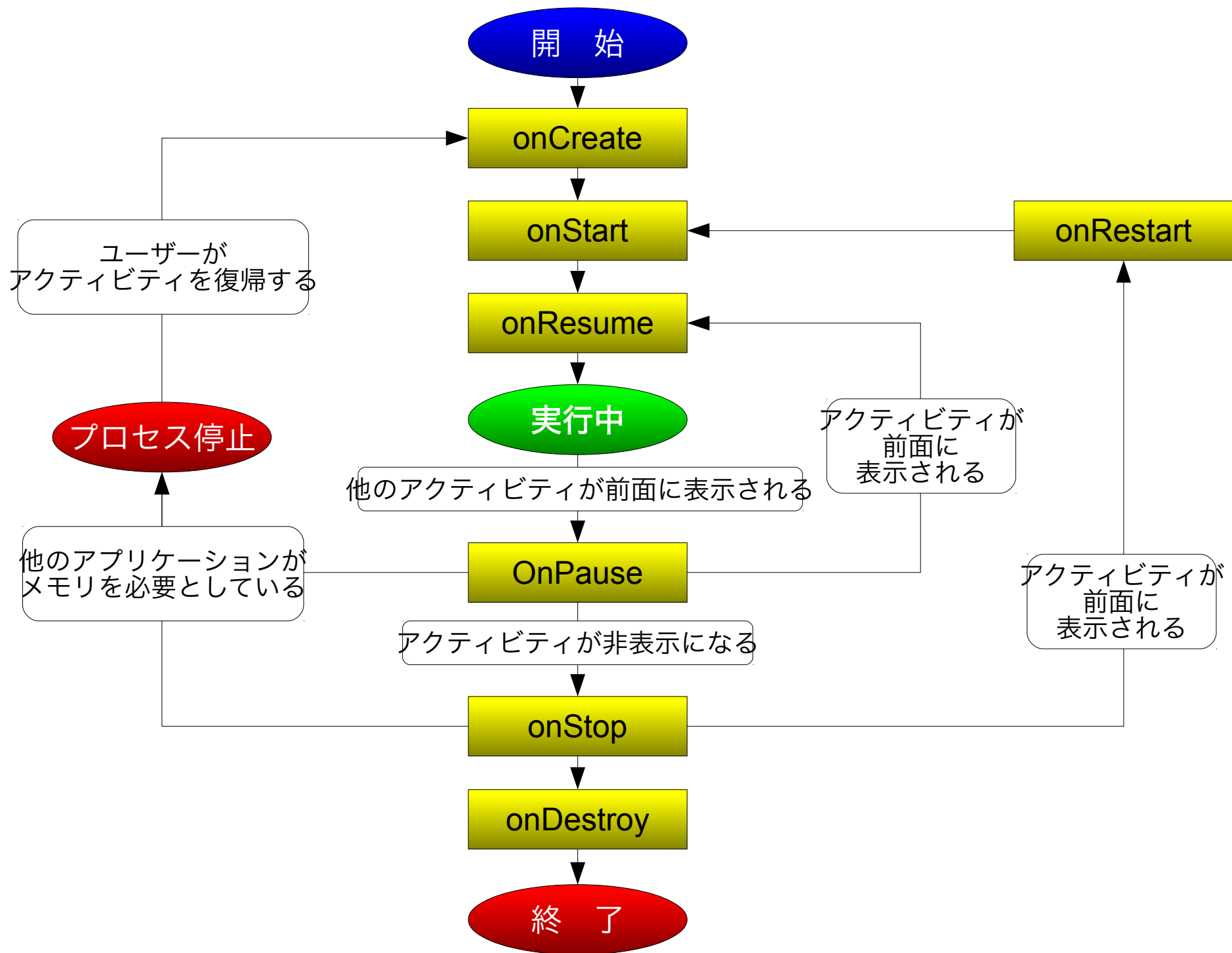
- 。 アプリケーションが、外部からの様々な割り込みに応じて、状態を遷移するための仕組み
 - ユーザーがキー操作でアプリケーションを終了した場合
 - アプリケーションを操作中に電話がかかってきた場合
 - ユーザーが一定時間操作をせず、システムがスリープの状態に移行した場合

■ アクティビティのライフサイクル



Labyrinth Lite





■ AndroidのUI

- Androidの画面は、XMLで設計する。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Direct input"
    />
<Button android:id="@+id/main_btn_test"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello2"
    />
</LinearLayout>
```

- 様々な標準部品(ビュー)を持っている
 - TextView - 文字列の表示
 - Button - ボタン

■ Androidのビュー

TextView



ImageView



ZoomButton

EditText

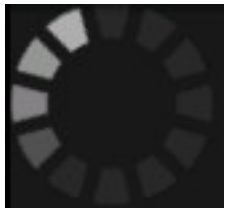
Button

Spinner



ImageButton

■ Androidのビュー



ProgressBar



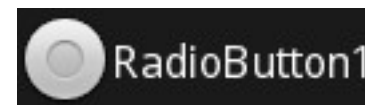
SeekBar



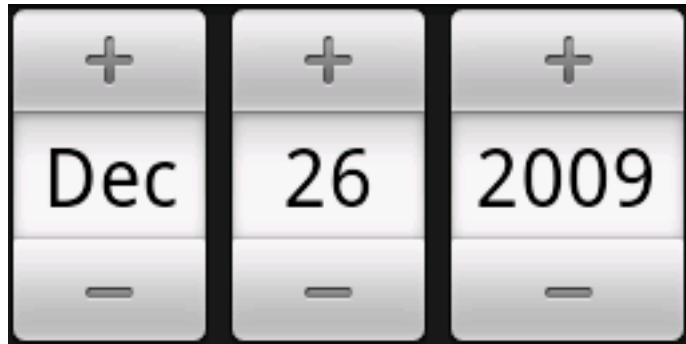
RatingBar



ToggleButton



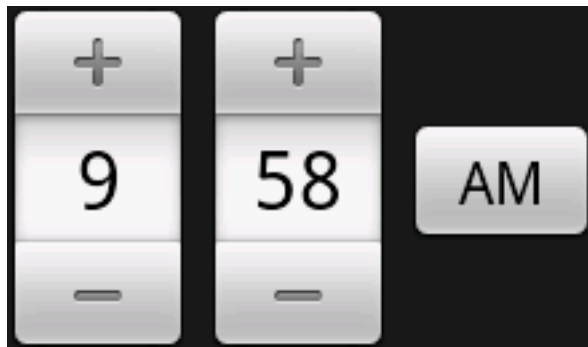
■ Androidのビュー



DatePicker



DigitalClock



TimePicker



AnalogClock

■ Layout

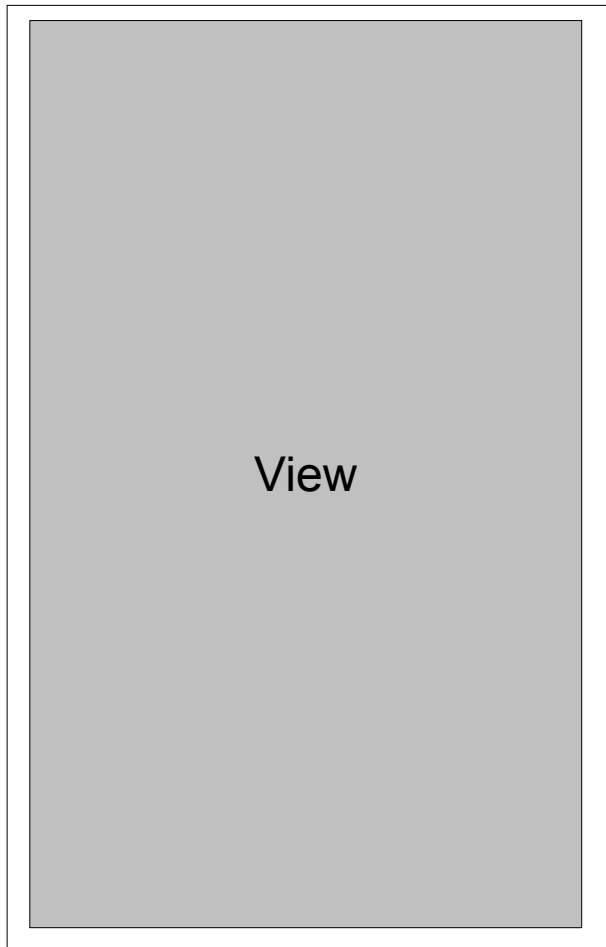
- Androidのウィジットを格納するコンテナ

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Direct input"
    />
<Button android:id="@+id/main_btn_test"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello2"
    />
</LinearLayout>
```

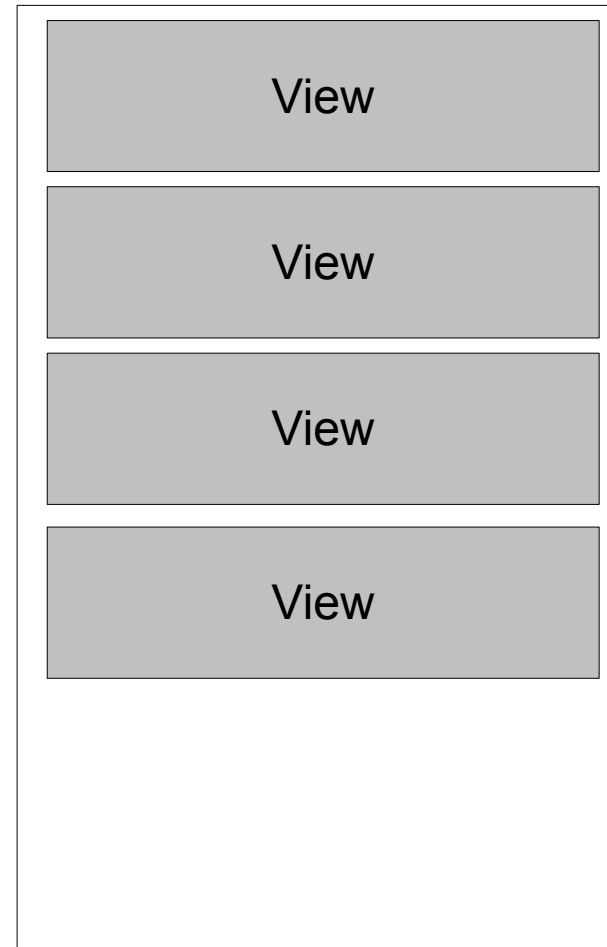
- コンテナによって並べ方に特徴がある
 - LinearLayoutは、内部のビューを並べて（この場合は縦に）表示する

■ Androidのレイアウト

- FrameLayout

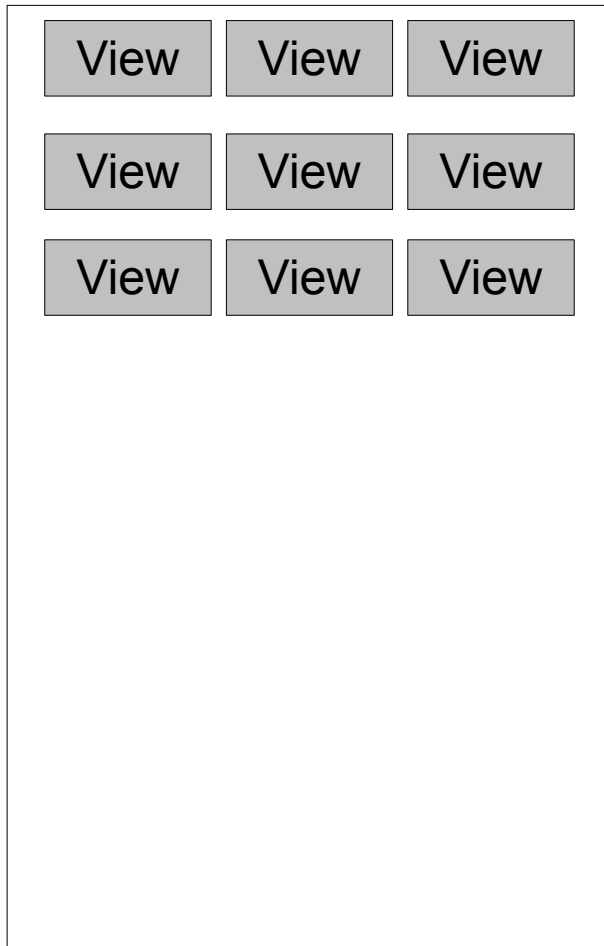


- LinearLayout

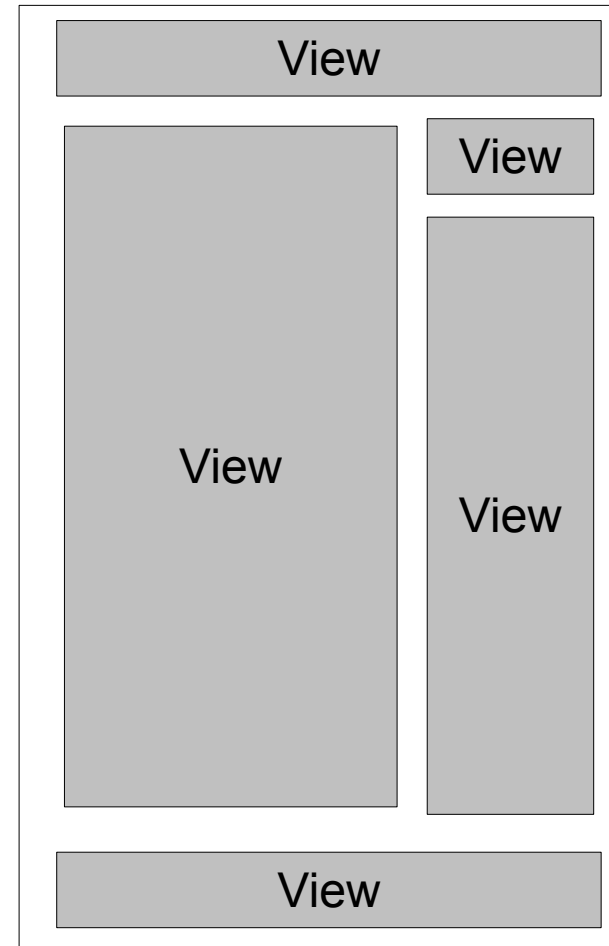


■ Androidのレイアウト

- TableLayout

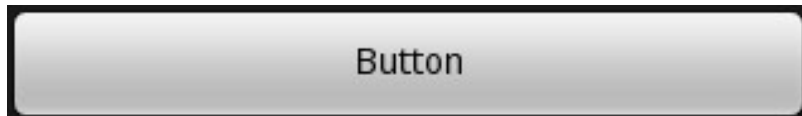


- RelativeLayout



■ カスタムボタンを作る

アプリデザインの都合上、
標準のボタンからデザインを変更したい



■ 画像素材を準備する



https://github.com/keiji/HandsOn-JAG-Hamamatsu-20110813/blob/master/img/button_normal.png

res/drawable/ に追加する

■ カスタムする三つの手段

- Viewクラスを継承した部品を 1 から作成する
- Buttonクラスを継承した新しいクラスを作成する
- Buttonタグの属性値で設定する

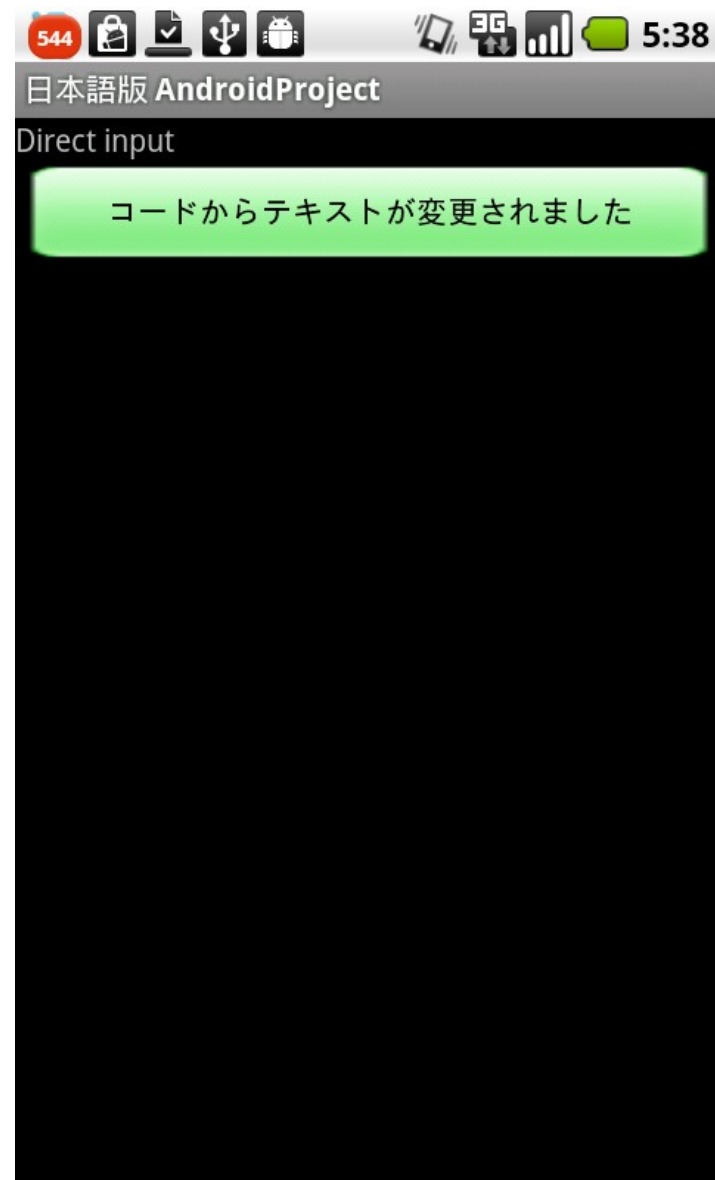
■ Buttonに適用する

- res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Direct input"
    />
<Button android:id="@+id/main_btn_test"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello2"
    android:background="@drawable/button_normal"
    />
</LinearLayout>
```

追加

■ 実行



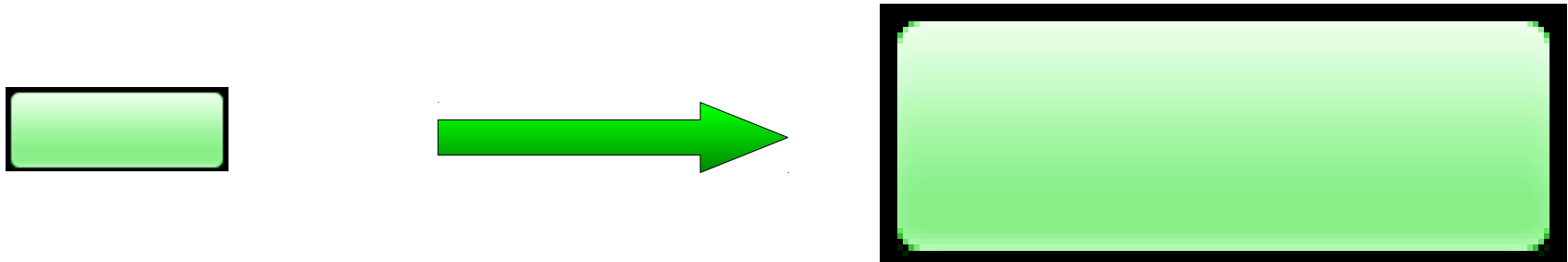
■ 画像の拡大に伴う崩れ

- 。 画像の自動拡大に伴い、角が横方向に間延びしている



■ 9-patch とは

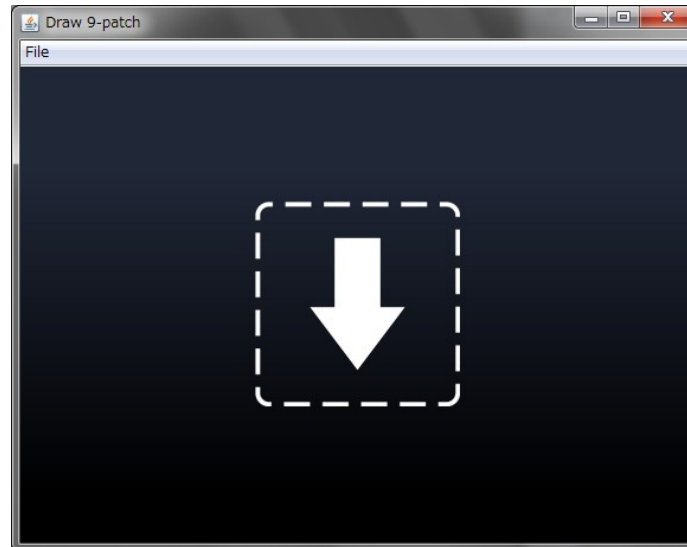
- 。 画像の拡大処理をスムーズに行う手法
- 。 通常の拡大処理では、



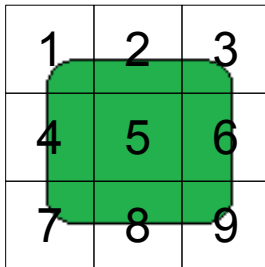
普通に拡大すると、荒さが目立つ

■ Draw 9-patch ツール

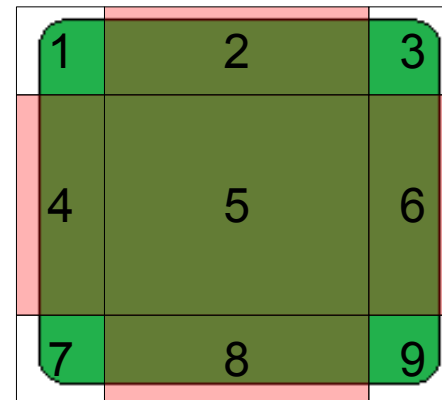
- Android SDKに含まれるツール
 - [SDKディレクトリ]\tools\draw9patch.bat
- 9-patch 画像を簡単に作成できる
 - 画像を9つに分割して、別々に保存する必要が無い



■ 9-patchとは？



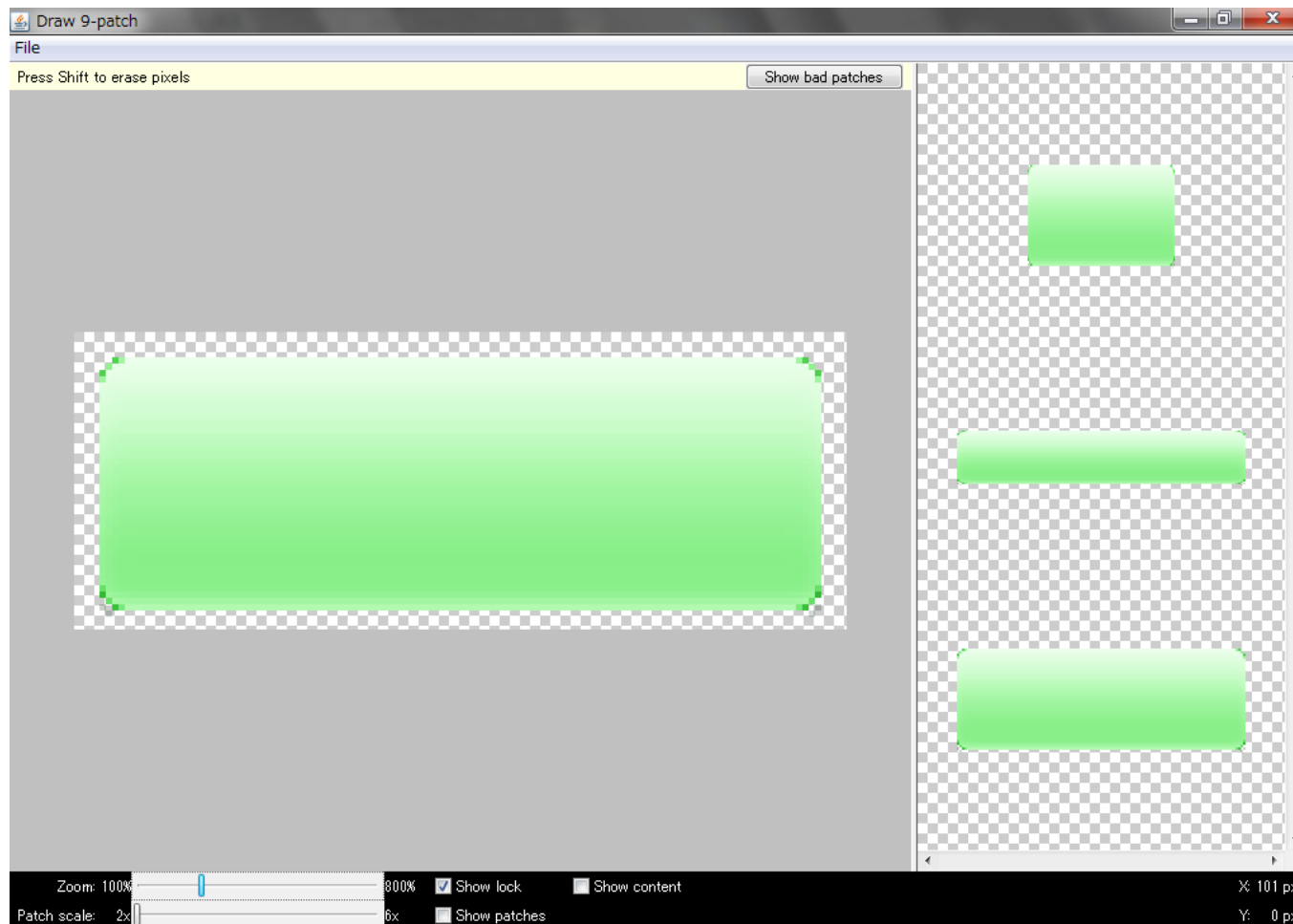
一つの画像を9つに分割
別々に保存。



1, 3, 7, 9の角は拡大せず
赤色部分(2, 4, 5, 6, 8)を
引き延ばす

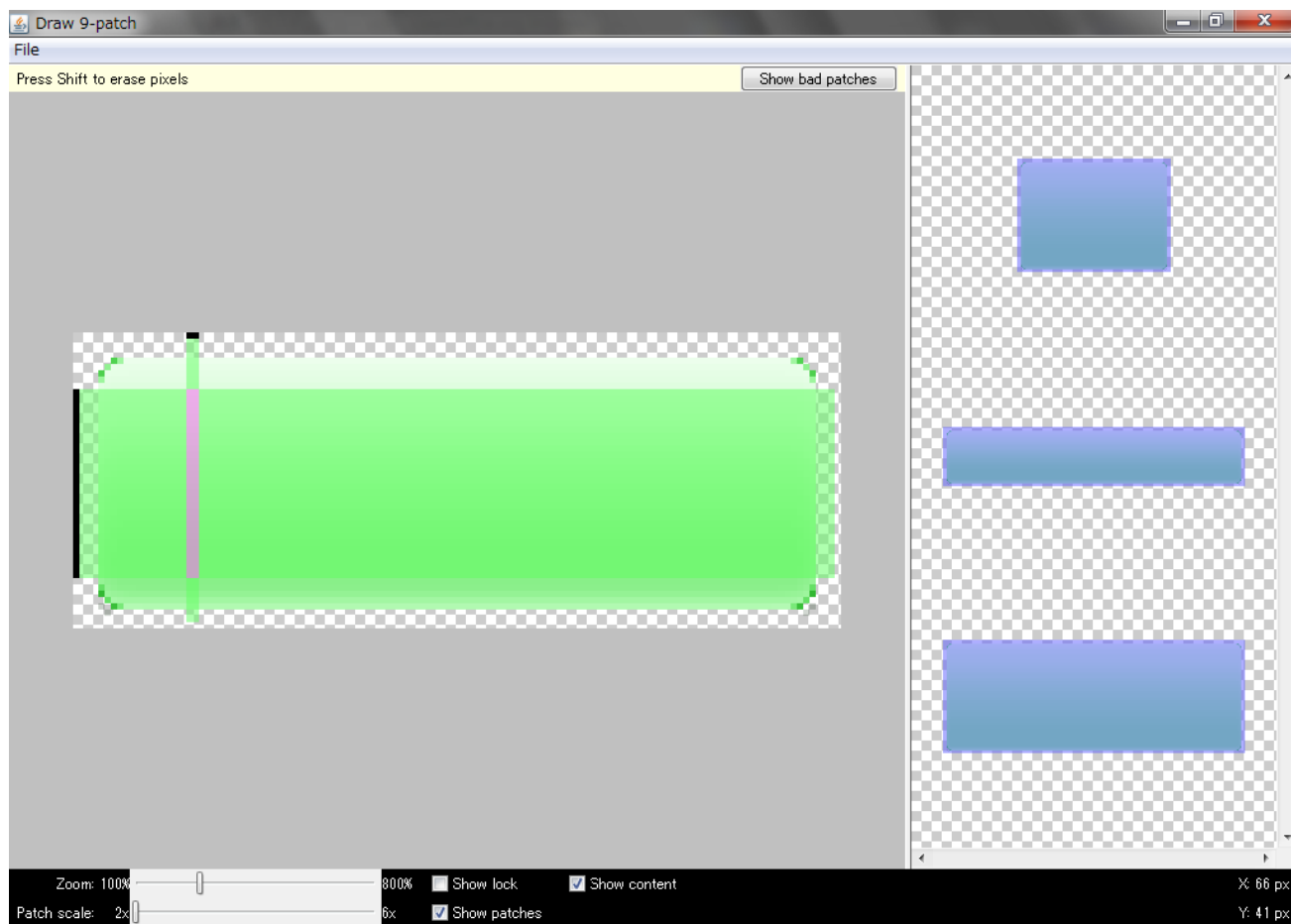
■ Draw 9-patch ツールの使い方

読み込み直後。右ペインには、それぞれ
縦方向への拡大、横方向への拡大、
縦横均等に拡大した場合の、プレビューが表示されている



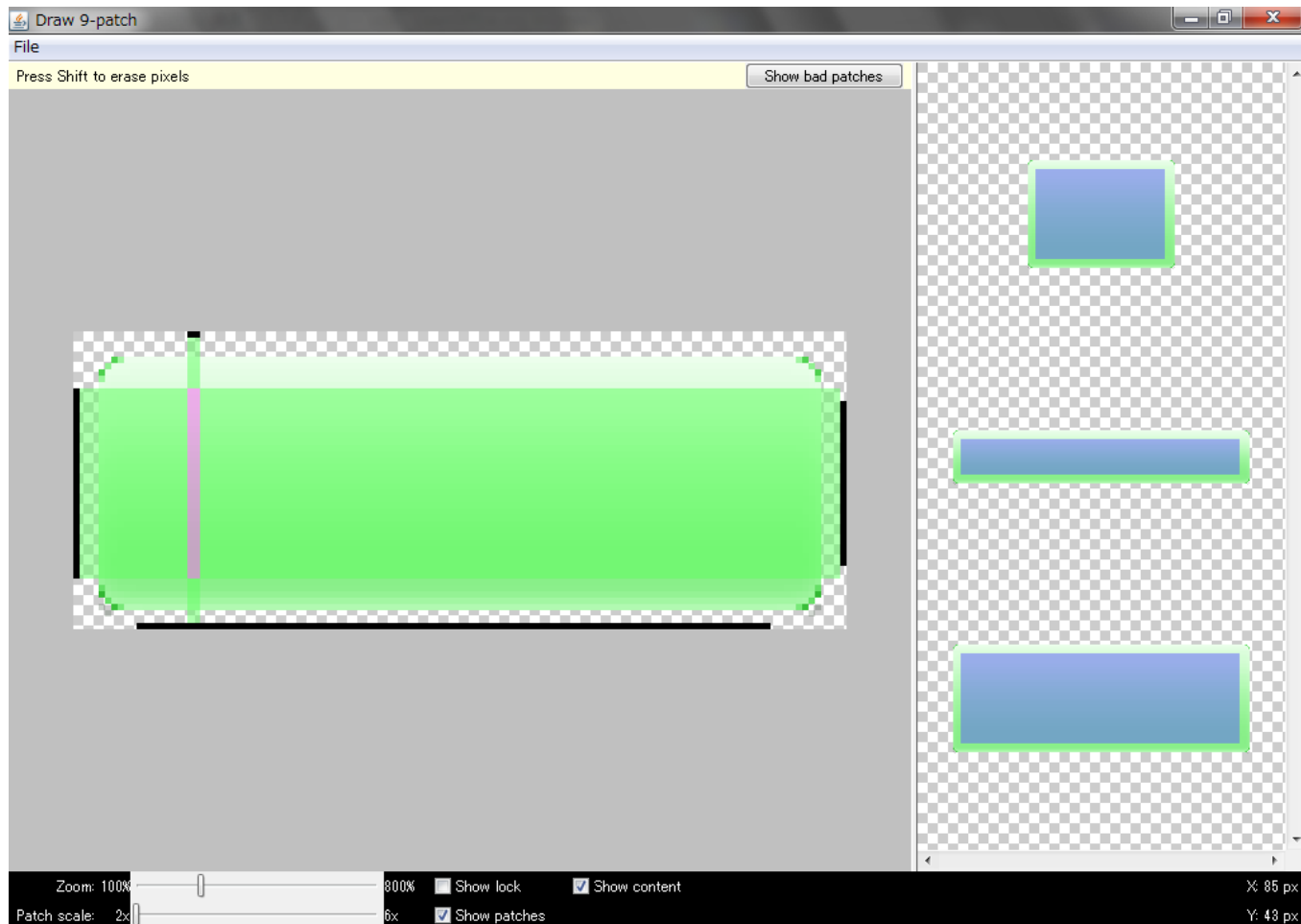
■ Draw 9-patch ツールの使い方

上と左に、クリックで1ピクセル分の黒ラインを引く。
黒ラインの領域が拡大、縮小時に引き延ばされるピクセル。 下と右は、ボタ
ン内に文字を描画するコンテンツ領域を指定している



■ Draw 9-patch ツールの使い方

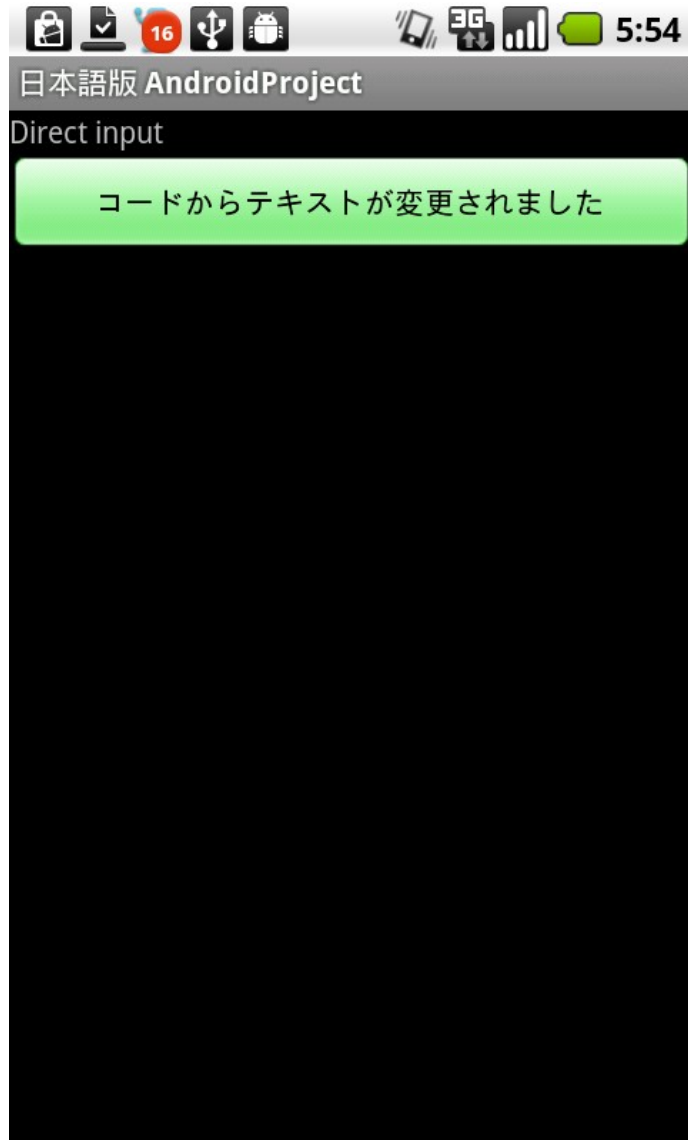
TextView等の背景画像に使う場合などは、画像のどの領域に文字列(コンテンツ)を表示するか指定する必要がある。
画像の右端と下側に黒ラインを引くことで、コンテンツの表示位置を明示的に指定できる



■ 画像を入れ替え

- res/drawable/button_normal.png を、削除する
- res/drawable/button_normal.9.png を、追加する

■ 実行



コードからテキストが変更されました

9-patch設定無し

コードからテキストが変更されました

9-patch設定有り

■ ボタンをタップしたときに反応したい

- ユーザーがボタンを押したときに、
「押されたとわかるよう」に変化させたい
 - ユーザーインタラクション



■ 画像の用意

- 。 押している時に表示する（切り替える）画像を用意する



https://github.com/keiji/HandsOn-JAG-Hamamatsu-20110813/blob/master/img/button_pressed.png

9-patchを設定してres/drawable/ に追加する

■ Selectorを設定する

- drawable/button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true"
        android:drawable="@drawable/button_pressed" />
    <item
        android:drawable="@drawable/button_normal" />
</selector>
```

- ビューの状態によって、表示内容を動的に変化する条件を設定する

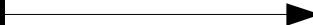
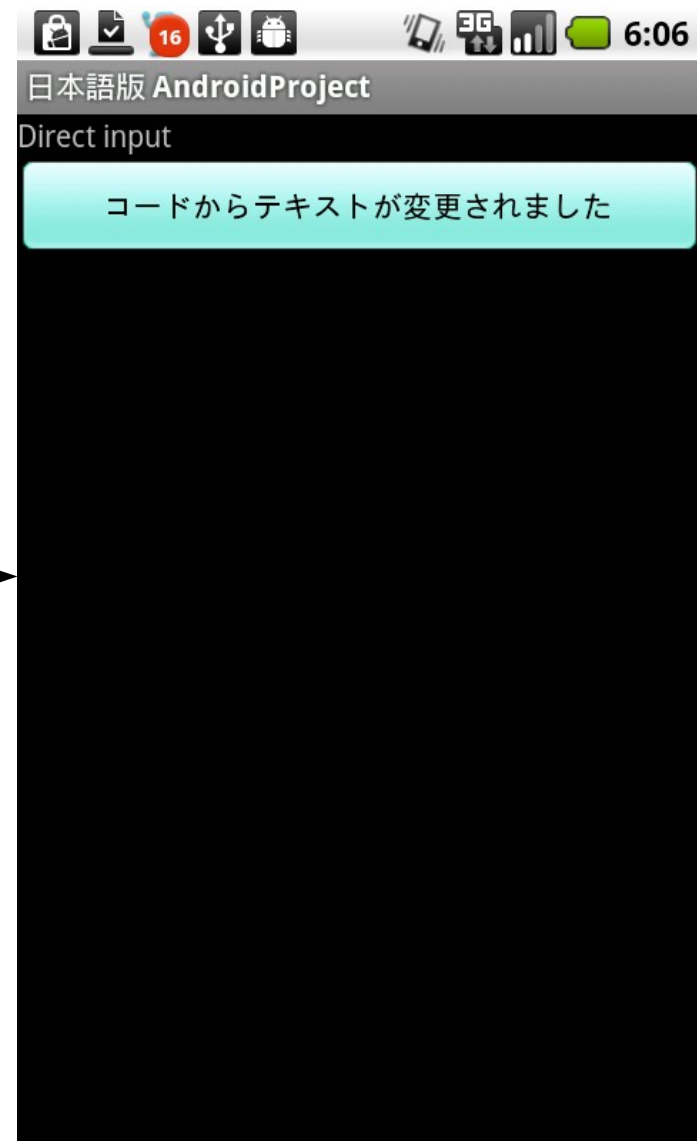
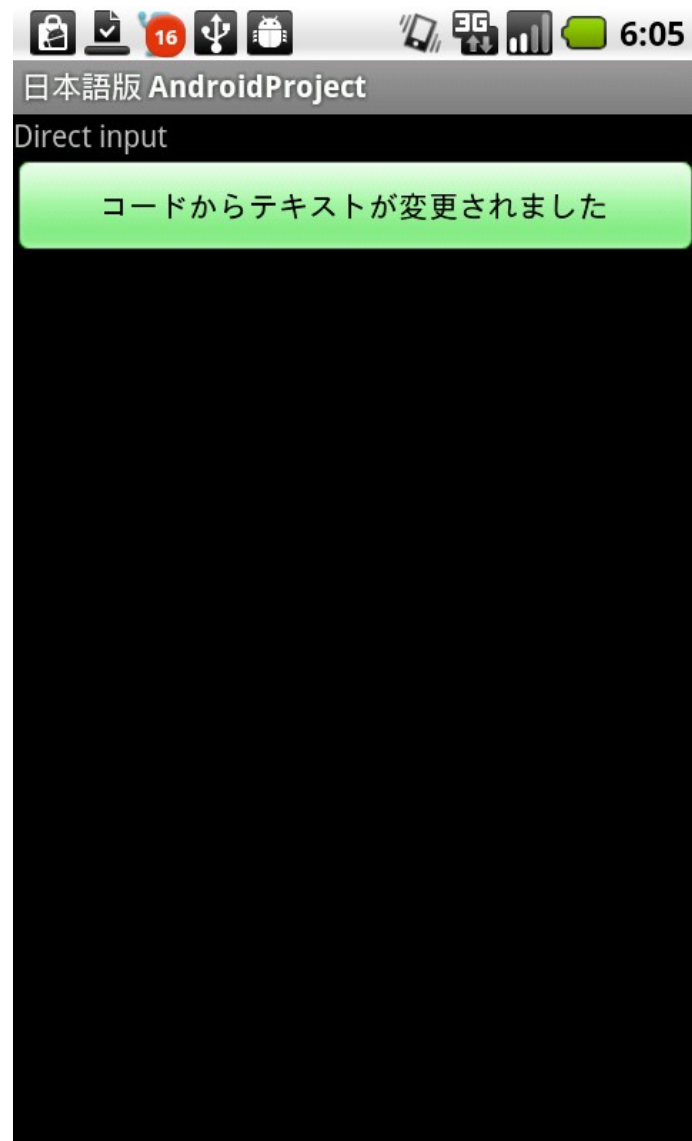
■ 参照先をSelectorに変更

- res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Direct input"
        />
    <Button android:id="@+id/main_btn_test"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello2"
        android:background="@drawable/button"
        />
</LinearLayout>
```

変更

■ 実行



■ カスタムUI作成のコツ 1

- 公開されているAndroidのソースコードを読む
 - オープンソース万歳！
- カスタムUIの多くは、画面の部品を1から作ることなく、実現できる
- 出来ること・出来ないことの境界を知る
 - 例えば、XMLによる属性指定でプログラマ的な挙動を変えることは、原則としてできない

■ 参考

- Androidデフォルトのボタンの画像リソース



btn_default_normal_disable.9.png



btn_default_normal.9.png



btn_default_normal_disable_focused.9.png



btn_default_pressed.9.png



btn_default_selected.9.png

■ 参考

。 Android標準のボタンのセレクター

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2008 The Android Open Source Project -->
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_window_focused="false"
        android:state_enabled="true"
        android:drawable="@drawable/btn_default_normal" />
    <item android:state_window_focused="false"
        android:state_enabled="false"
        android:drawable="@drawable/btn_default_normal_disable" />
    <item android:state_pressed="true"
        android:drawable="@drawable/btn_default_pressed" />
    <item android:state_focused="true" android:state_enabled="true"
        android:drawable="@drawable/btn_default_selected" />
    <item android:state_enabled="true"
        android:drawable="@drawable/btn_default_normal" />
    <item android:state_focused="true"
        android:drawable="@drawable/btn_default_normal_disable_focused" />
    <item
        android:drawable="@drawable/btn_default_normal_disable" />
</selector>
```

質疑応答

本資料は、[有限会社シリーズ](#)の著作物であり、
クリエイティブコモンズの表示-非営利-継承 3.0 Unported ライセンスの元で公開しています。



本資料の内容の一部は、Googleが作成、提供しているコンテンツを複製したもので、
クリエイティブコモンズの表示 2.5 ライセンスに記載の条件に従って使用しています。

