

2015 年 10 月 29 日 (木) 実施

レイアウト

レイアウトの位置付け

Android アプリで用いられる様々なレイアウト (Layout) は、それぞれ ViewGroup クラスを継承するクラスとして定義されているものであり、ViewGroup クラスは View クラスを継承するクラスである。View クラスはユーザインタフェースを構成する基本要素を表す。ビュー (View) は画面上に長方形の領域を占め、描画及びイベント操作を担う。ビューグループ (ViewGroup) は他のビューを含むことが出来る、不可視のコンテナ (容器) である。

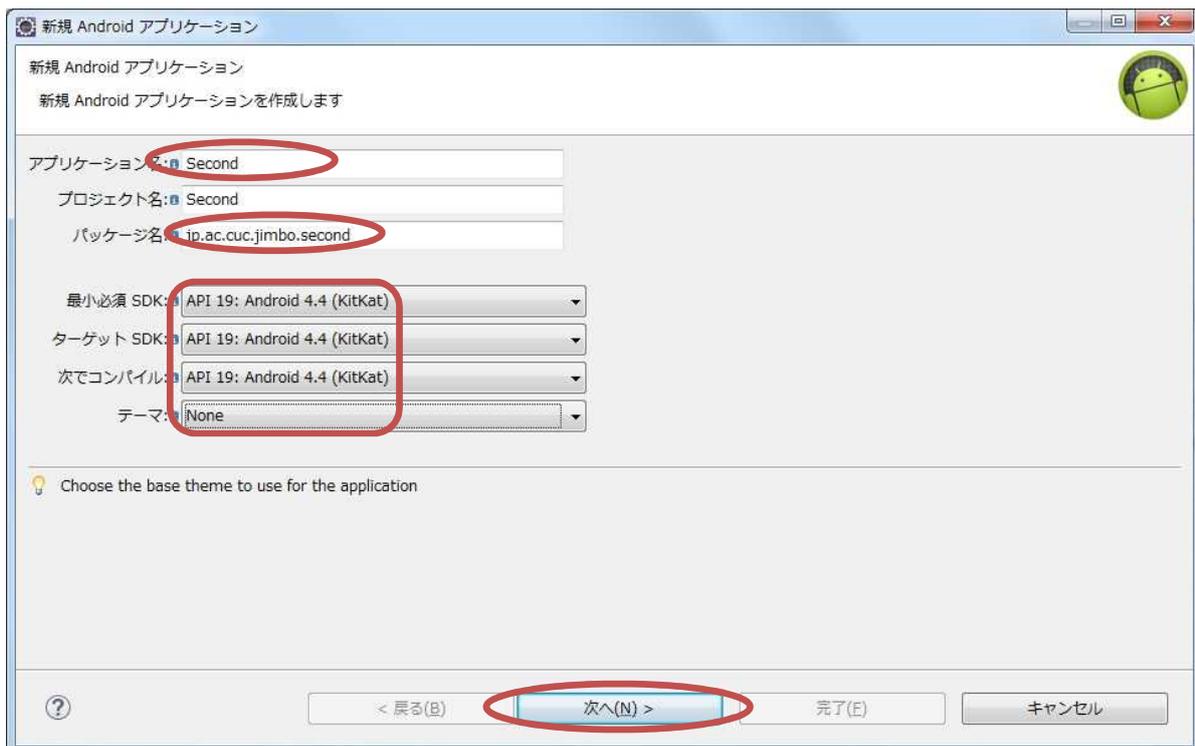
課題

今回は、新規のプロジェクトを作成し、これまで用いて来た RelativeLayout 以外のレイアウトとして、LinearLayout, GridLayout 及び TableLayout を用いた画面定義を作成し、それぞれのレイアウトの特性を学ぶ。

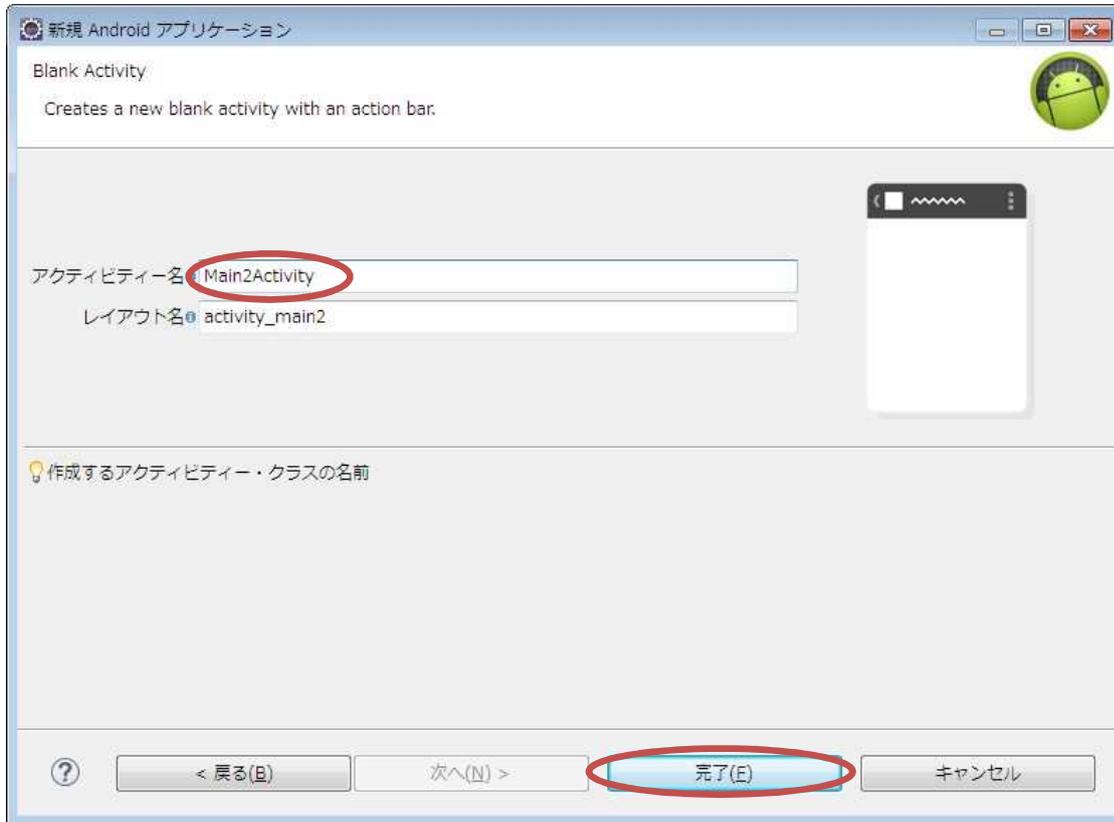
Android アプリの作成

Eclipse を起動し、『ファイル』→『新規』→『Android アプリケーション・プロジェクト』と選択する。

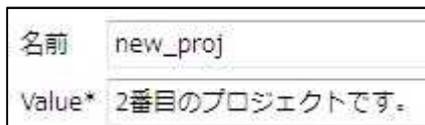
『アプリケーション名』を「**Second**」、『パッケージ名』を「**jp. ac. cuc. jimbo. second**」(下線部は自分の名前に置き換える)、『最小必須 SDK』、『ターゲット SDK』及び『次でコンパイル』を「API 19」とし、『テーマ』を「None」とする。(今回用いるレイアウトの中に Android SDK の版が API 14 以降でないと対応していないものがある)



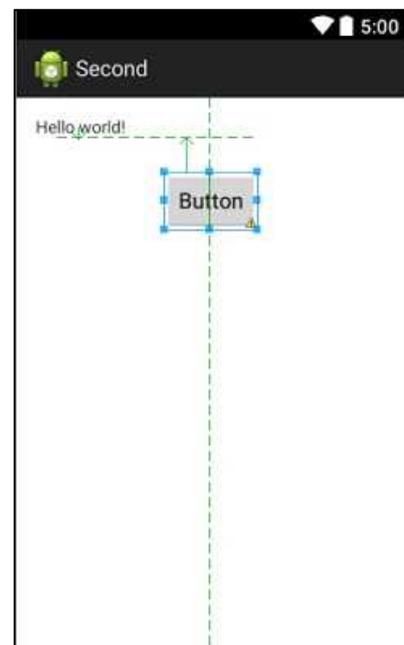
『カスタム・ランチャー・アイコンを作成する』のチェックを外して、『次へ』をクリックする。
 『Blank Activity』を選択して、『次へ』をクリックする。
 『アクティビティ名』を「Main2Activity」に変更して、『完了』をクリックする。



『res』 → 『values』 と展開し、『strings.xml』を開き、『追加』をクリックする。『String』を選択して OK をクリックし、『名前』を「new_proj」, 『Value』を「2 番目のプロジェクトです。」と入力する。同様に、『名前』:「button_label」, 『Value』:「レイアウト表示」の String を追加し、保管して閉じる。



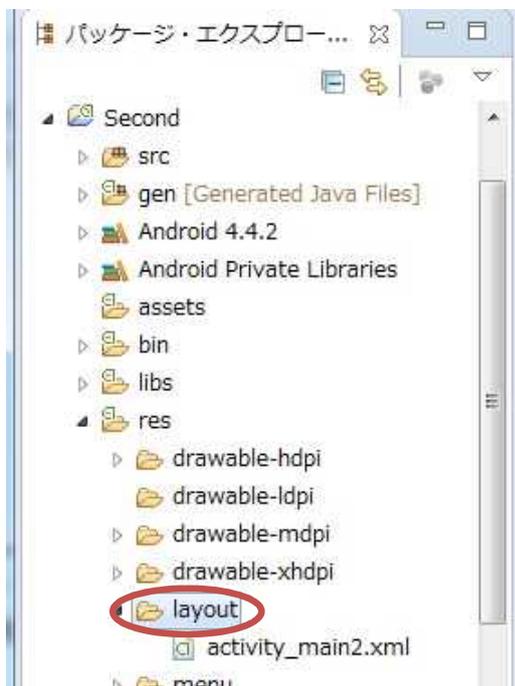
activity_main2.xml にボタンを配置する。



ボタンの Text には『button_label』, テキストビューの Text には『new_proj』を設定する。

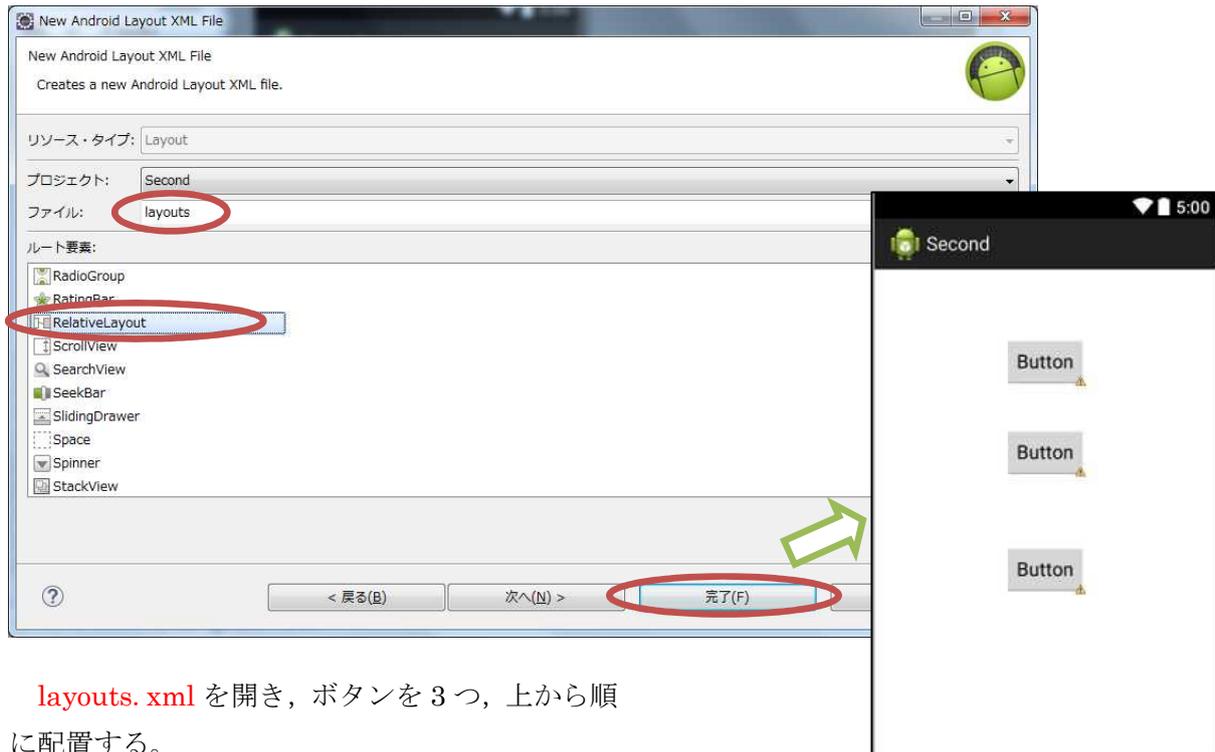


パッケージ・エクスプローラーの『Second』→『res』を展開して『layout』を選択し、『ファイル』→『新規』→『その他』と選択する。



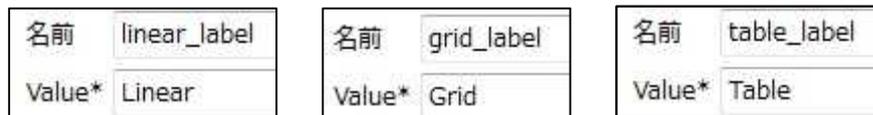
『Android』を展開し、『Android XML レイアウト・ファイル』を選択し、『次へ』をクリックする。『ファイル』の欄には「**layouts**」と入力し、『RelativeLayout』を選択して、『完了』をクリックする。

(図は次のページ)

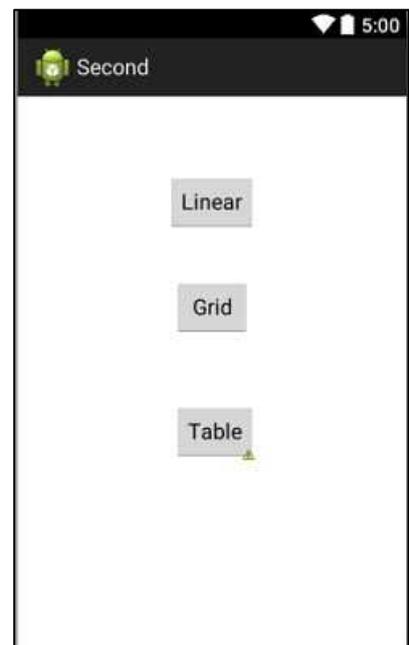


layouts.xml を開き、ボタンを 3 つ、上から順に配置する。

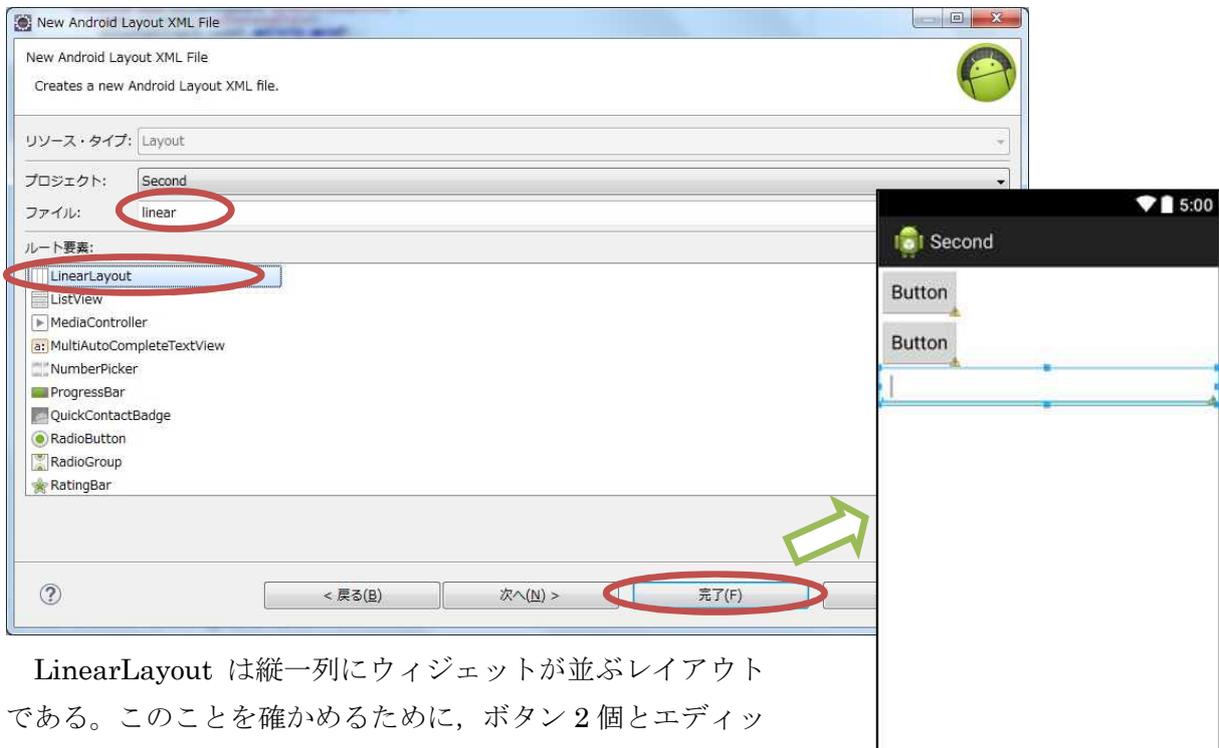
『res』 → 『values』 と展開し、『strings.xml』を開き、『追加』をクリックする。『String』を選択して OK をクリックし、『名前』を「linear_label」、『Value』を「Linear」と入力する。同様に、『名前』:「grid_label」、『Value』:「Grid」、『名前』:「table_label」、『Value』:「Table」の 2 つの String を追加する。



一番上の『Button』(button1)を選択し、『Properties』にある『Text』の右側の『・・・』ボタンをクリックする。『linear_label』を選択し、OK をクリックする。他のウィジェットも右図の様に設定する。『保管』のアイコンをクリックし、layouts.xml を上書き保存して閉じる。

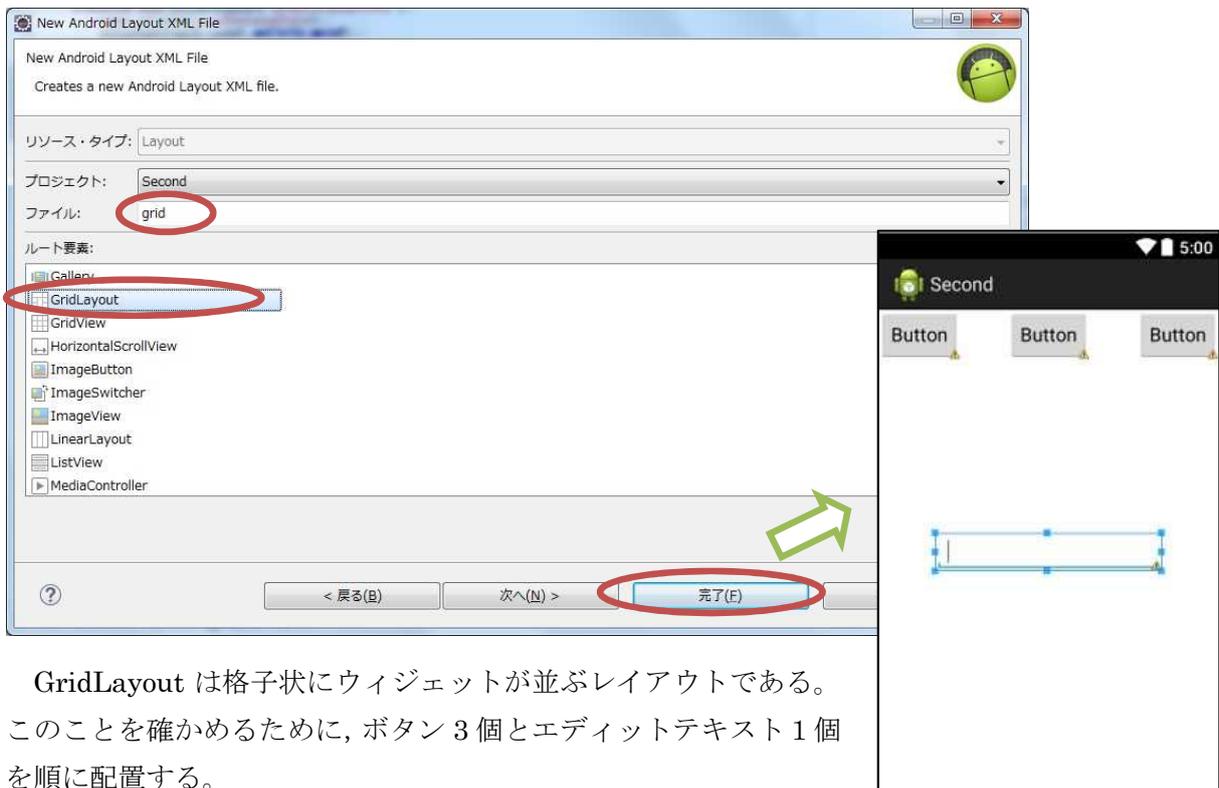


『同様に、『Android XML レイアウト・ファイル』を新規作成する。『ファイル』の欄には「linear」と入力し、『LinearLayout』を選択して、『完了』をクリックする。



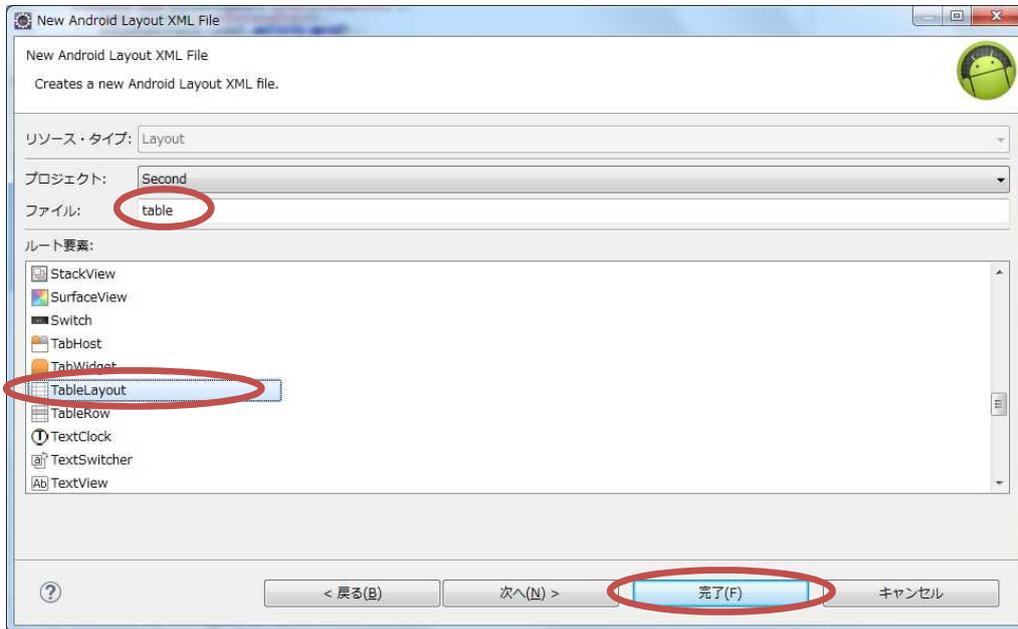
LinearLayout は縦一列にウィジェットが並ぶレイアウトである。このことを確かめるために、ボタン 2 個とエディットテキスト 1 個を順に配置する。

同様に、『Android XML レイアウト・ファイル』を新規作成する。『ファイル』の欄には「grid」と入力し、『GridLayout』を選択して、『完了』をクリックする。

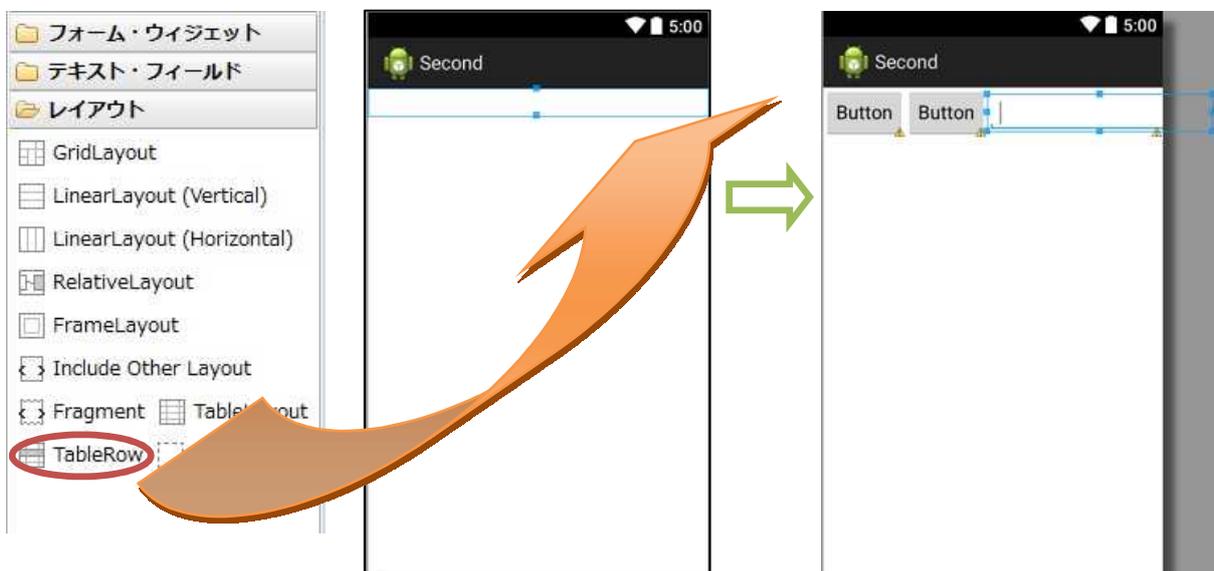


GridLayout は格子状にウィジェットが並ぶレイアウトである。このことを確かめるために、ボタン 3 個とエディットテキスト 1 個を順に配置する。

同様に、『Android XML レイアウト・ファイル』を新規作成する。『ファイル』の欄には「table」と入力し、『TableLayout』を選択して、『完了』をクリックする。



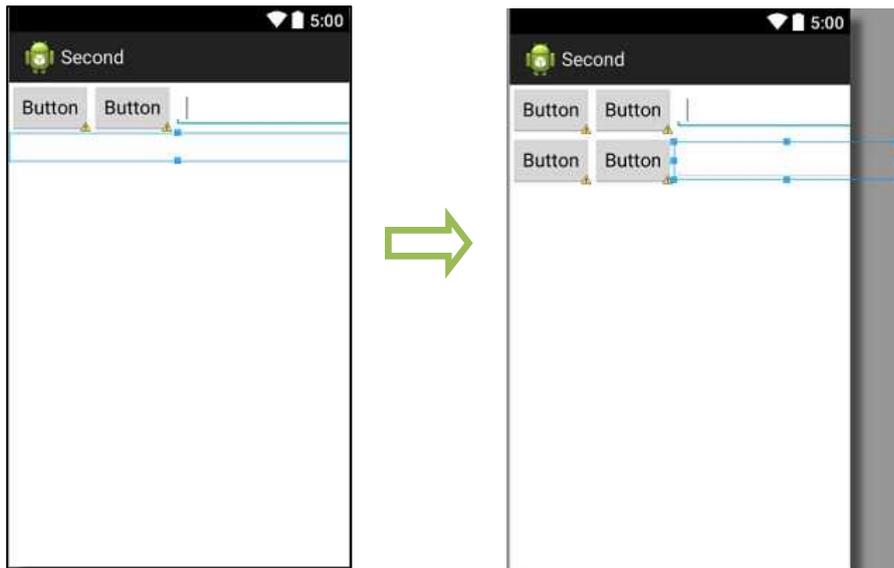
TableLayout は、TableRow を配置して、その中にウィジェットを配置するレイアウトである。このことを確かめるために、先ず TableRow を『レイアウト』からドラッグして配置し、次にその中にボタン 2 個とエディットテキスト 1 個を順に配置する。



* 実用的なアプリでは、エディットテキストの幅を適切に設定すべきであるが、ここではそのまま貼った場合の「wrap_content」のままにしている。

更に、TableRow を配置して、その中にボタン 2 個とエディットテキスト 1 個を順に配置する。

(図は次のページ)



Main2Activity.java の onCreate メソッドの中に Button ウィジェット **button1** のインスタンス **btn** を作成し、それに働きかけるイベントリスナーを付け加える。

```
Button btn = (Button)this.findViewById(R.id.button1);
btn.setOnClickListener(
    new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            drawLayouts();
        }
    }
);
```

```
Main2Activity.java
1 package jp.ac.cuc.jimbo.second;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Menu;
6 import android.view.MenuItem;
7 import android.view.View;
8 import android.widget.Button;
9
10 public class Main2Activity extends Activity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main2);
16         Button btn = (Button)this.findViewById(R.id.button1);
17         btn.setOnClickListener(
18             new View.OnClickListener() {
19
20                 @Override
21                 public void onClick(View v) {
22                     drawLayouts();
23                 }
24             }
25         );
26     }
27 }
```

この中で呼び出している `drawLayouts` メソッドは `onCreate` メソッドの下に自作するメソッドであり、その中でレイアウトを `layouts.xml` に設定し直し、そこで用いられている `Button` ウィジェット `button1~3` のインスタンス `btn1~3` を作成し、それらに働きかけるイベントリスナーを記述する。それぞれのイベントリスナーには、対応するレイアウトを引数とする `setContentView` メソッドを記述する。

```
public void drawLayouts() {
    setContentView(R.layout.layouts);
    Button btn1 = (Button) this.findViewById(R.id.button1);
    btn1.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                setContentView(R.layout.linear);
            }
        }
    );

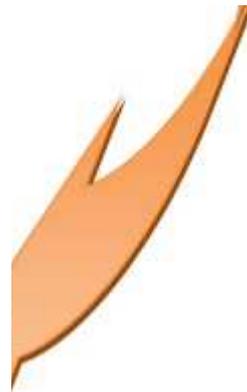
    Button btn2 = (Button) this.findViewById(R.id.button2);
    btn2.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                setContentView(R.layout.grid);
            }
        }
    );

    Button btn3 = (Button) this.findViewById(R.id.button3);
    btn3.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                setContentView(R.layout.table);
            }
        }
    );
}
```

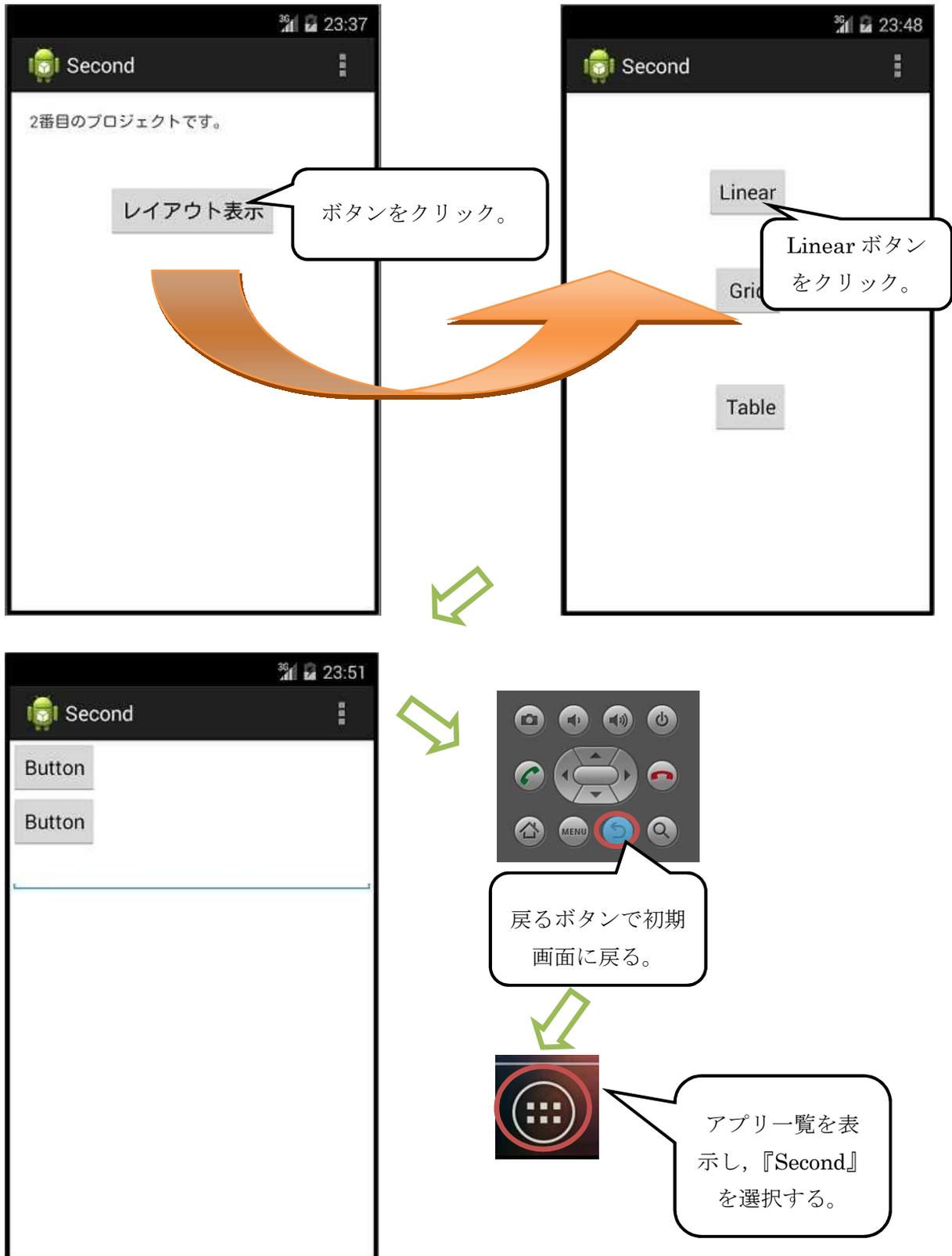
```

Main2Activity.java
20     > > > @Override
21     > > > public void onClick(View v) {
22     > > >     drawLayouts();
23     > > > }
24     > > }
25     > > );
26     > > }
27
28     > public void drawLayouts() {
29     > > setContentView(R.layout.layouts);
30     > > Button btn1 = (Button)this.findViewById(R.id.button1);
31     > > btn1.setOnClickListener(
32     > >     new View.OnClickListener() {
33     > > >
34     > > > @Override
35     > > > public void onClick(View v) {
36     > > > > setContentView(R.layout.linear);
37     > > > > }
38     > > > > }
39     > > > );
40
41     > > Button btn2 = (Button)this.findViewById(R.id.button2);
42     > > btn2.setOnClickListener(
43     > >     new View.OnClickListener() {
44     > > >
45     > > > @Override
46     > > > public void onClick(View v) {
47     > > > > setContentView(R.layout.grid);
48     > > > > }
49     > > > > }
50     > > > );
51
52     > > Button btn3 = (Button)this.findViewById(R.id.button3);
53     > > btn3.setOnClickListener(
54     > >     new View.OnClickListener() {
55     > > >
56     > > > @Override
57     > > > public void onClick(View v) {
58     > > > > setContentView(R.layout.table);
59     > > > > }
60     > > > > }
61     > > > );
62     > > }
63
64     > @Override
65     > public boolean onCreateOptionsMenu(Menu menu) {

```

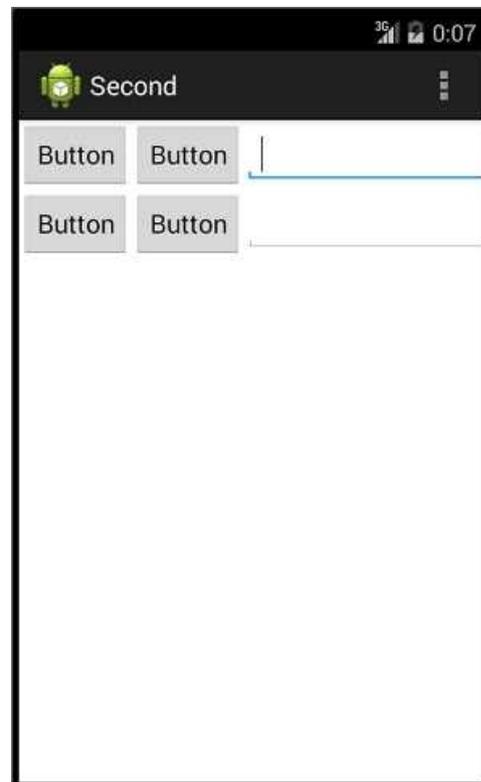
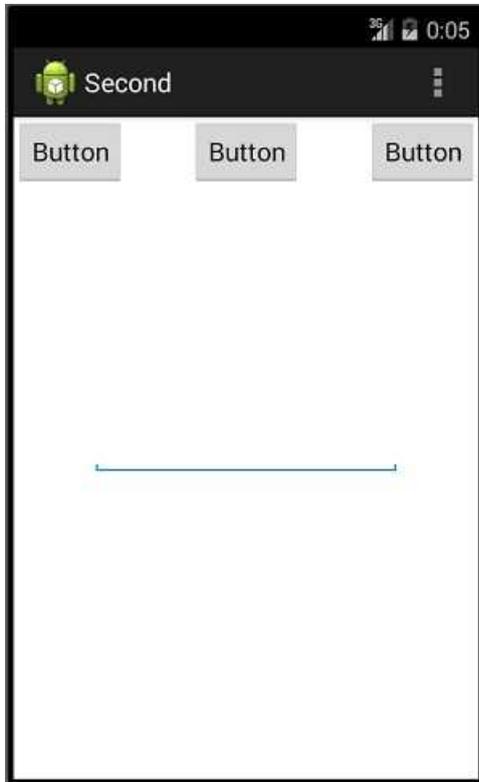


『保管』のアイコンをクリックして、Main2Activity.java を上書き保存し、パッケージ・エクスプローラーの『Second』を選択して、実行ボタンをクリックする。



初期画面に戻ったら、ボタンをクリックして layouts の画面に移り、Grid ボタンをクリックする。(下左図)

画面が切り替わったら、戻るボタンで初期画面に戻り、アプリ一覧から『Second』を選択し、ボタンをクリックして layouts の画面に移り、Table ボタンをクリックする。(下右図)



提出物 :

- 1) 画面のレイアウト設定ファイル **linear.xml**
- 2) 画面のレイアウト設定ファイル **grid.xml**
- 3) 画面のレイアウト設定ファイル **table.xml**
- 4) 画面のレイアウト設定ファイル **layouts.xml**
- 5) 画面のレイアウト設定ファイル **activity_main2.xml**
- 6) Main2Activity のソースファイル **Main2Activity.java**