2015 年 12 月 3 日（木）実施

効果音の付加

SoundPool とは

Android には音を処理するクラスが複数用意されているが，その中で SoundPool は，予め音のデータをメモリ上に読み込んで再生するため，長い音楽よりも短い音を扱うのに適している。また，SoundPool では遅延が無いので，効果音を付加したい場面で用いられる。

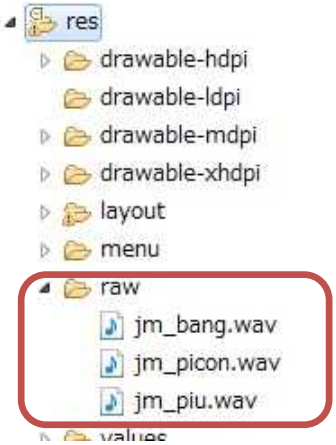
課 題

今回は，様々なアクティビティから SoundPool を利用可能なクラスを作成し，効果音の付加の基本を学ぶ。

Android アプリの作成

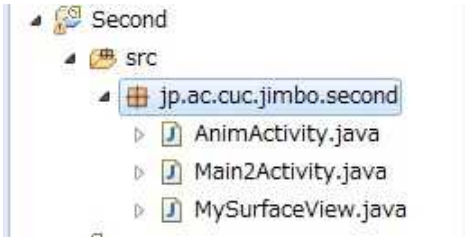
H:¥workspace¥Second¥res で，「raw」という名前のフォルダを作成し，ダウンロードした音データファイル（フリーソフトウェア『SWave』で授業担当者が自作したもの）をコピーして貼り付ける。

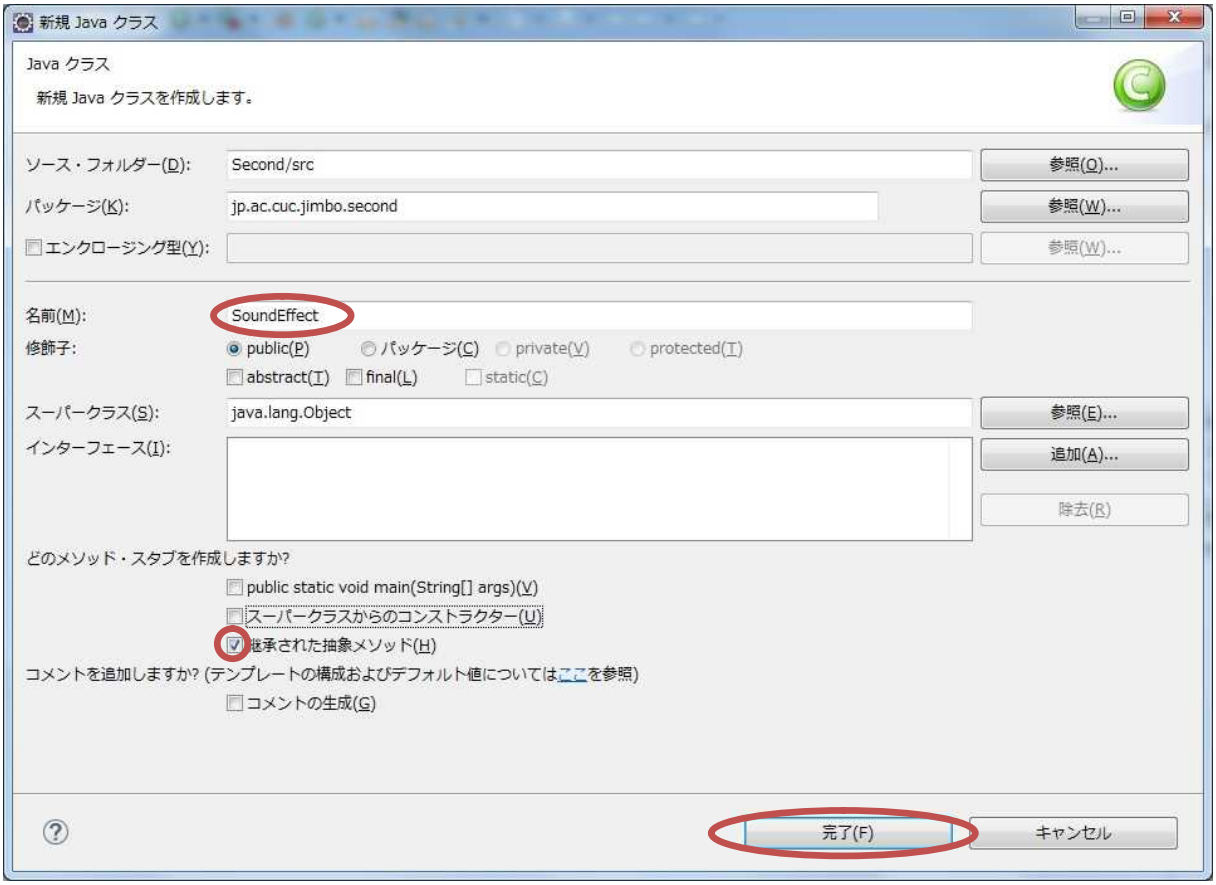
Eclipse を起動し，パッケージ・エクスプローラーでこれらが見えていることを確認する。（先に起動していた場合には，リフレッシュを適用する。）



パッケージ・エクスプローラーでパッケージ名を選択し，『ファイル』→『新規』→『クラス』と選択して，『次へ』をクリックする。

『名前』を「SoundEffect」とする。





SoundEffect. java に次の内容を付け加える。（適宜，インポートの編成を行う。）

```
private static final int[] soundRes = {R.raw.jm_piu, R.raw.jm_picon, R.raw.jm_bang};
private SoundPool soundPool;
private int[] soundID = new int[soundRes.length];
boolean loaded = false;

public SoundEffect(Context context) {
    soundPool = new SoundPool(1, AudioManager.STREAM_MUSIC, 0);
    soundPool.setOnLoadCompleteListener(new OnLoadCompleteListener() {
        @Override
        public void onLoadComplete(SoundPool soundPool, int sampleId, int status){
            loaded = true;
        }
    });
```

```
    for (int i = 0; i < soundRes.length; i++)
        soundID[i] = soundPool.load(context, soundRes[i], 1);
}

public void destroy() {
    for (int i = 0; i < soundRes.length; i++)
        soundPool.unload(soundRes[i]);

    soundPool.release();
}

public void playSoundEffect(int i) {
    int loop;

    if (i == 1)
        loop = 2;
    else
        loop = 0;

    if (loaded)
        soundPool.play(soundID[i], 1.0f, 1.0f, 1, loop, 1.0f);
}
```

SoundEffect.java

```
package jp.ac.cuc.jimbo.second;

import android.content.Context;
import android.media.AudioManager;
import android.media.SoundPool;
import android.media.SoundPool.OnLoadCompleteListener;

public class SoundEffect {
    private static final int[] soundRes = {R.raw.jm_piu, R.raw.jm_picon, R.raw.jm_bang};
    private SoundPool soundPool;
    private int[] soundID = new int[soundRes.length];
    boolean loaded = false;

    public SoundEffect(Context context) {
        soundPool = new SoundPool(1, AudioManager.STREAM_MUSIC, 0);
        soundPool.setOnLoadCompleteListener(new OnLoadCompleteListener() {
            @Override
            public void onLoadComplete(SoundPool soundPool, int sampleId, int status){
                loaded = true;
            }
        });

        for (int i = 0; i < soundRes.length; i++)
            soundID[i] = soundPool.load(context, soundRes[i], 1);
    }

    public void destroy() {
        for (int i = 0; i < soundRes.length; i++)
            soundPool.unload(soundRes[i]);

        soundPool.release();
    }

    public void playSoundEffect(int i) {
        int loop;

        if (i == 1)
            loop = 2;
        else
            loop = 0;

        if (loaded)
            soundPool.play(soundID[i], 1.0f, 1.0f, 1, loop, 1.0f);
    }
}
```

＊ インポートの編成のオプションでは，android.media.SoundPool.OnLoadCompleteListener を選択する。

==【説明】==========

フィールド

soundRes・・・リソースにある音データ　ここでは要素 3 個（soundRes[0]，soundRes[1]，soundRes[2] ）の配列

soundPool・・・SoundPool クラスのインスタンス

soundID・・・load メソッドで，音データをメモリ上に読み込んだ際に割り当てられる整数　ここでは要素 3 個（soundID [0]，soundID [1]，soundID [2] ）の配列

loaded・・・音データの読み込みが完了した際に真理値を true にセットする。(初期値は false)

コンストラクタ　SoundEffect(Context context)

SoundPool(1，AudioManager. STREAM_MUSIC，0)・・・SoundPool クラスのインスタンスを生成する際，SoundPool クラスのコンストラクタは，第 1 引数には同時に再生する音データの最大数，第 2 引数には音データのタイプ（通常は STREAM_MUSIC)，第 3 引数にはサンプルレートコンバータの品質（デフォルト値は 0）を与えて初期化する。

setOnLoadCompleteListener メソッド・・・OnLoadCompleteListener インターフェイスからのコールバックを受け止める。

onLoadComplete メソッド・・・音データの読み込みが完了すると呼ばれる。

load(context，soundRes[i]，1)・・・音データを読み込むための SoundPool クラスのメソッド　第 1 引数にはコンテキスト（アプリの属性)，第 2 引数にはリソースの ID，第 3 引数には音の優先順位（現状ではダミー，将来的な互換性のために 1 としておく）を指定する。

destroy()メソッド

unload(soundRes[i])・・・メモリ上に読み込んでいた音データのリソースを解放するための SoundPool クラスのメソッド

release()・・・SoundPool クラスのインスタンスが利用していたメモリの領域全体を解放するための SoundPool クラスのメソッド

playSoundEffect(int i)

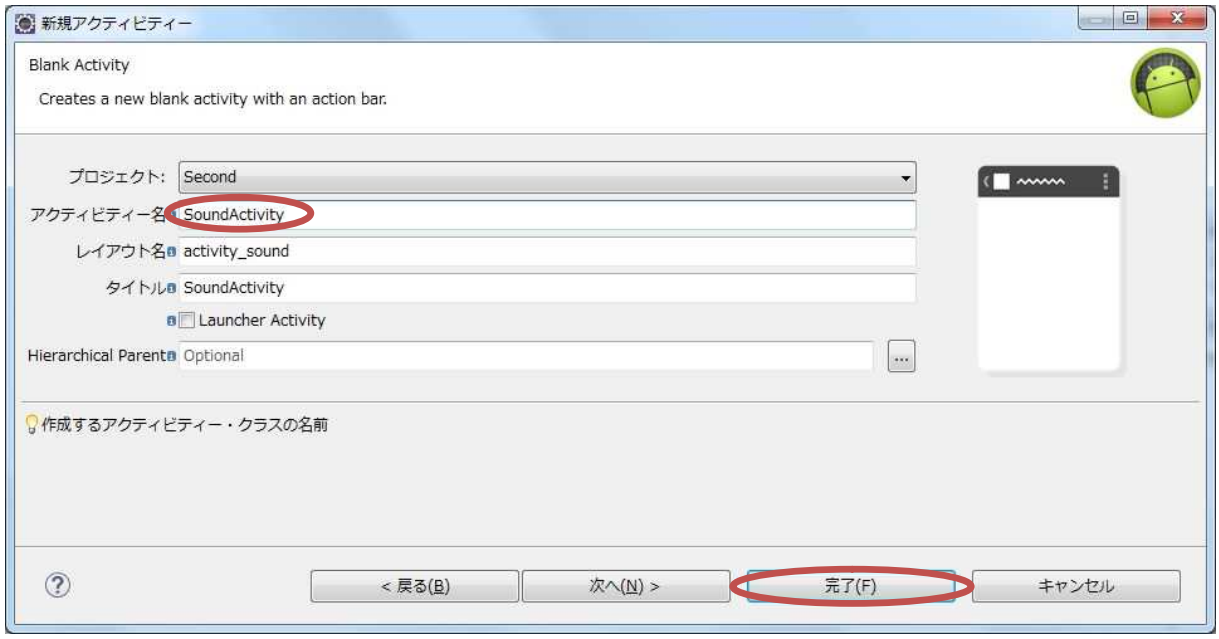
play(soundID[i]，1. 0f，1. 0f，1，loop，1. 0f)・・・音データを再生するための SoundPool クラスのメソッド　第 1 引数には load メソッドで返される soundID，第 2 引数には左側のボリューム（0. 0～1. 0)，第 3 引数には右側のボリューム（0. 0～1. 0)，第 4 引数には音の優先順位，第 5 引数には繰り返しの数（0 は繰り返しなし，-1 は無限ループ)，第 6 引数には再生レート（0. 5～2. 0，標準は 1）を指定する。

==========

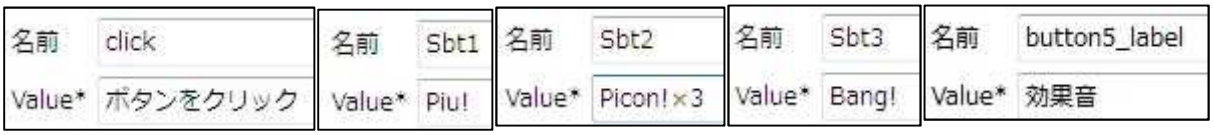
パッケージ・エクスプローラーでパッケージ名を選択し，『ファイル』→『新規』→『Android アクティビティー』と選択して，『次へ』をクリックする。

『Blank Activity』を選択して，『次へ』をクリックする。

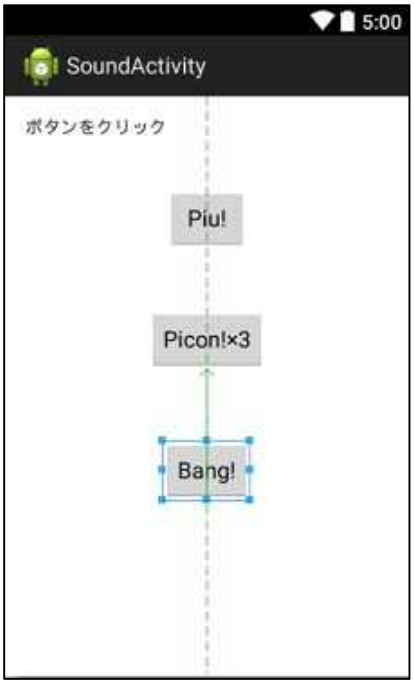
『アクティビティー名』を「SoundActivity」に書き換え，『完了』をクリックする。



『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「click」，『Value』を「ボタンをクリック」と入力する。同様にして，『名前』：「Sbt1」，『Value』：「Piu!」；『名前』：「Sbt2」，『Value』：「Picon!×3」；『名前』：「Sbt3」，『Value』：「Bang!」；『名前』：「button5_label」，『Value』：「効果音」を追加して，保管して閉じる。



activity_sound. xml でボタンを 3 個配置し，Text 属性に登録した文字列を設定する。



SoundActivity. java に色付きの枠の箇所を付け加える。（適宜，インポートの編成を行う。）

SoundActivity.java

```
1  package jp.ac.cuc.jimbo.second;
2
3  import android.app.Activity;
10
11 public class SoundActivity extends Activity {
12     public static final int SOUND_PIU = 0;
13     public static final int SOUND_PICON = 1;
14     public static final int SOUND_BANG = 2;
15     private SoundEffect soundEffect;
16
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_sound);
22         Context context = getApplicationContext();
23         soundEffect = new SoundEffect(context);
24
25         Button btn1 = (Button)this.findViewById(R.id.button1);
26         btn1.setOnClickListener(
27             new View.OnClickListener() {
28
29                 @Override
30                 public void onClick(View v) {
31                     soundEffect.playSoundEffect(SOUND_PIU);
32                 }
33             }
34         );
35
36         Button btn2 = (Button)this.findViewById(R.id.button2);
37         btn2.setOnClickListener(
38             new View.OnClickListener() {
39
40                 @Override
41                 public void onClick(View v) {
42                     soundEffect.playSoundEffect(SOUND_PICON);
43                 }
44             }
45         );
46
47         Button btn3 = (Button)this.findViewById(R.id.button3);
48         btn3.setOnClickListener(
49             new View.OnClickListener() {
50
51                 @Override
52                 public void onClick(View v) {
53                     soundEffect.playSoundEffect(SOUND_BANG);
54                 }
55             }
56         );
57
58     }
59
60     @Override
61     protected void onPause() {
62         super.onPause();
63         soundEffect.destroy();
64     }
65
66     @Override
67     public boolean onCreateOptionsMenu(Menu menu) {
```

それぞれの色の枠内を次に示す。（ダウンロードしたファイル SoundActivity. txt を利用する）

```
public static final int SOUND_PIU = 0;
public static final int SOUND_PICON = 1;
public static final int SOUND_BANG = 2;
private SoundEffect soundEffect;
```

```
    Context context = getApplicationContext();
    soundEffect = new SoundEffect(context);

    Button btn1 = (Button)this.findViewById(R.id.button1);
    btn1.setOnClickListener(
        new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                soundEffect.playSoundEffect(SOUND_PIU);
            }
        }
    );

    Button btn2 = (Button)this.findViewById(R.id.button2);
    btn2.setOnClickListener(
        new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                soundEffect.playSoundEffect(SOUND_PICON);
            }
        }
    );

    Button btn3 = (Button)this.findViewById(R.id.button3);
    btn3.setOnClickListener(
        new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                soundEffect.playSoundEffect(SOUND_BANG);
            }
        }
    );
```

```
@Override
protected void onPause() {
    super.onPause();
    soundEffect.destroy();
}
```

==【説明】==========

1) 橙色の枠内　SoundActivity クラスのフィールド

SOUND_PIU, SOUND_PICON, SOUND_BANG・・・SoundEffect クラスの playSoundEffect メソッドの引数を分かり易くするために，それぞれの値を 0，1，2 に固定した変数

soundEffect・・・SoundEffect クラスのインスタンス

2) 赤色の枠内　onCreate メソッド内の記述

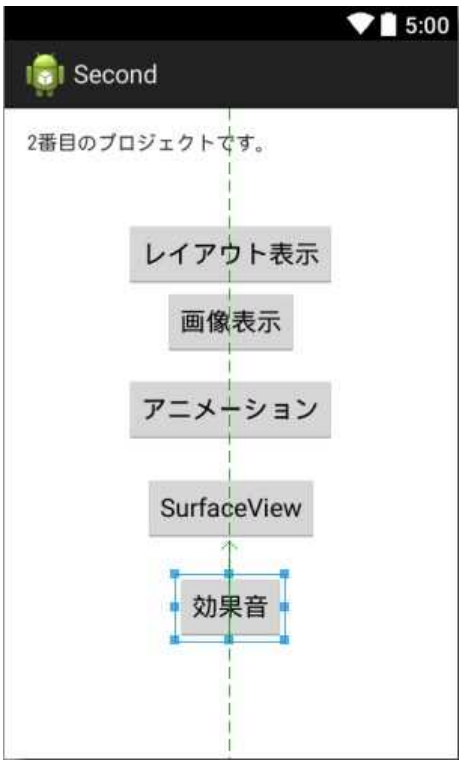
context・・・Context クラスのインスタンス

getApplicationContext()・・・コンテキストを取得して返すための Context クラスのメソッド

3) 水色の枠内　onPause メソッド・・・別のアクティビティが開始された時に呼び出されるメソッド　ここに SoundEffect クラスの destroy メソッドを適用し，リソースを解放する。

==========

activity_main2. xml にボタンを配置する。ボタンの Text には『button5_label』を設定する。



Main2Activity. java を開く。

onCreate メソッド中に Button ウィジェット button5 のインスタンス btn5 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。イベントリスナーでは，イベントリスナーには，インテントを記述する。(intent. setClassName の行は，教材の行幅の都合で折り返している。)

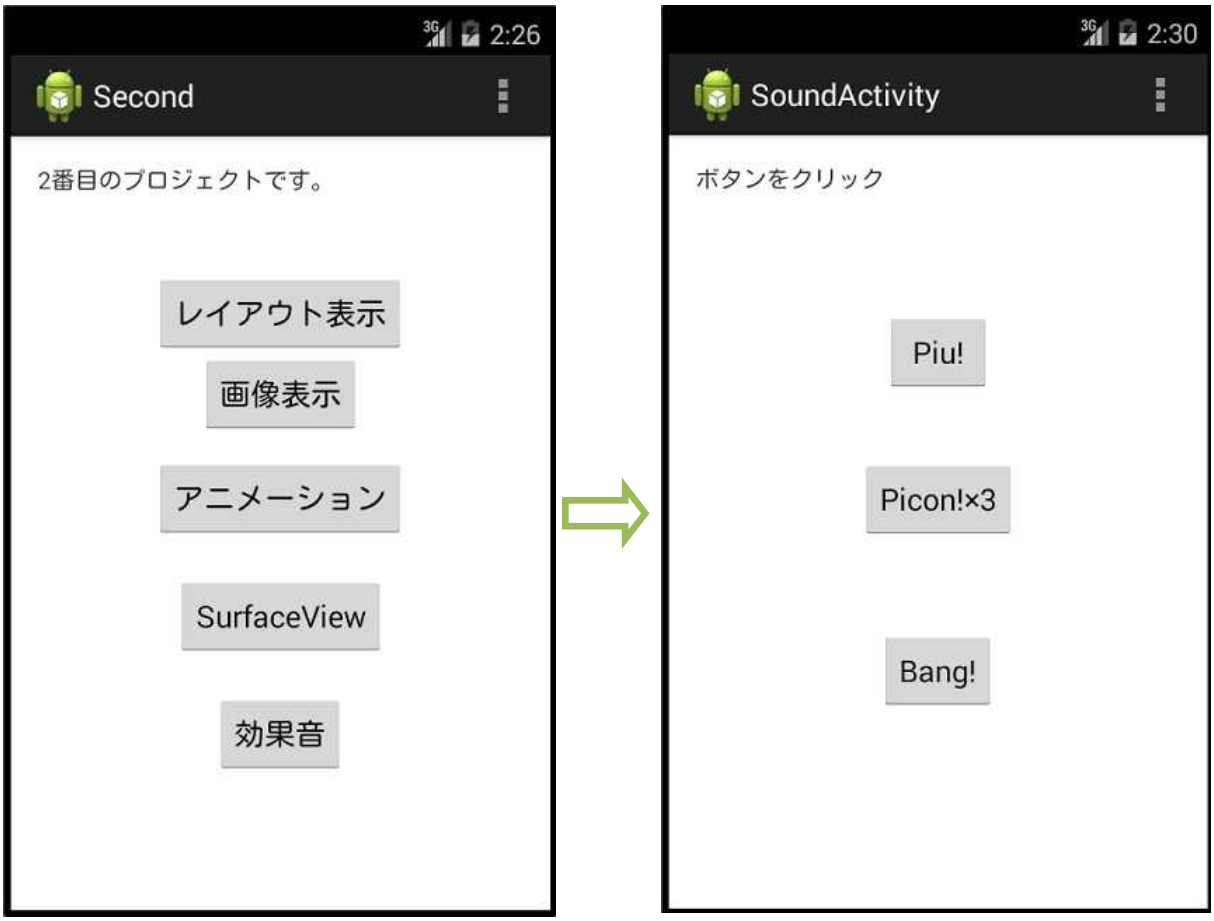
```
Button btn5 = (Button)this.findViewById(R.id.button5);
btn5.setOnClickListener(
    new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            Intent intent = new Intent();
            intent.setClassName("jp.ac.cuc.jimbo.second",
              "jp.ac.cuc.jimbo.second.SoundActivity");
            startActivity(intent);
        }
    }
);
```

Main2Activity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    Button btn = (Button)this.findViewById(R.id.button1);
    btn.setOnClickListener(
        new View.OnClickListener() {

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

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

            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.setClassName("jp.ac.cuc.jimbo.second", "jp.ac.cuc.jimbo.second.AnimActivity");
                startActivity(intent);
            }
        }
    );
    Button btn4 = (Button)this.findViewById(R.id.button4);
    btn4.setOnClickListener(
        new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                drawSurfaceView();
            }
        }
    );
    Button btn5 = (Button)this.findViewById(R.id.button5);
    btn5.setOnClickListener(
        new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.setClassName("jp.ac.cuc.jimbo.second", "jp.ac.cuc.jimbo.second.SoundActivity");
                startActivity(intent);
            }
        }
    );
}
```

『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・エクスプローラーで Second を選択して，実行ボタンをクリックする。

（図は次のページ）



提出物：

1）画面のレイアウト設定ファイル activity_sound. xml

2）画面のレイアウト設定ファイル activity_main2. xml

3）SoundEffect のソースファイル SoundEffect. java

4）Main2Activity のソースファイル Main2Activity. java