2016 年 12 月 8 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。

授業の準備

1）Android Studio の初期設定

Android Studio を起動し，『Configure』→『設定のインポート』を選択し，第 3 回の教材の p.5 に従って設定をインポートする。

2）プロジェクトの新規作成

『Application name』(アプリ名)を「Prog_9th」(先頭は大文字,「_」は下線),『Company Domain』を「b6a0xxx.cuc.ac.jp」に書き換え,『Project Location』の先頭の「C:¥Users¥ b6a0xxx」を『H:』に書き換えて,『次へ』ボタンを押す。

第 1 回と同様に『Minimum SDK』では『API 22』を選択する（第 1 回教材 p.7)。

『Activity name』は「MainActivity9」とする。

3）AVD の設定

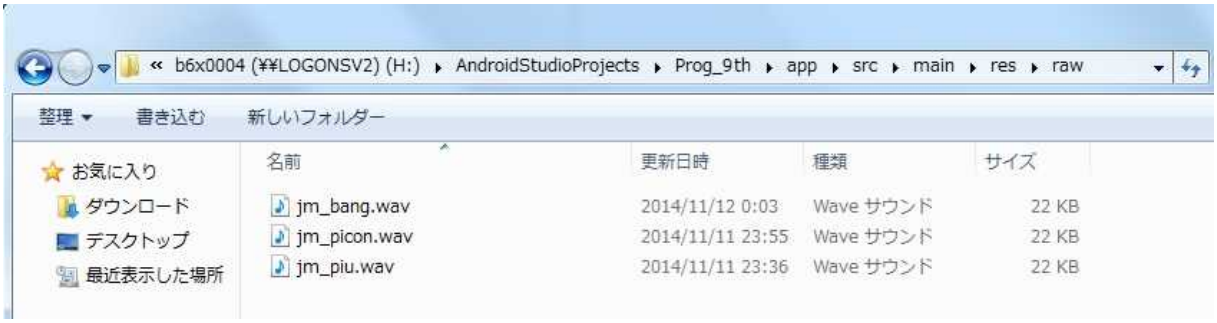
第 1 回の授業で作成した AVD の設定は H ドライブにあって残るが，SDK のシステムイメージは C ドライブにあるので，消失している。そこで,『Download』をクリックして，インストールし直す（第 2 回教材 p.4)。

課 題

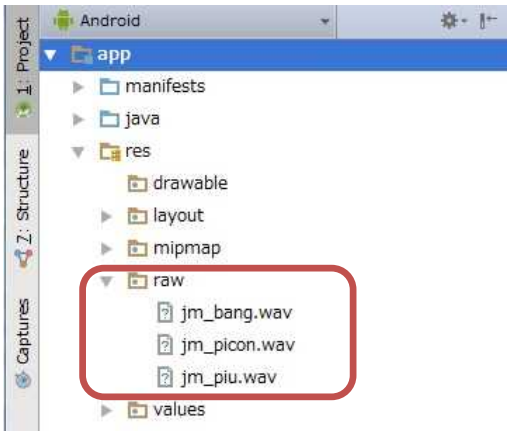
今回は，様々なアクティビティから SoundPool を利用可能なクラスを作成し，効果音の付加の基本を学ぶ。

Android アプリの作成

H:¥AndroidStudioProjects¥Prog_9th¥app¥src¥main¥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。

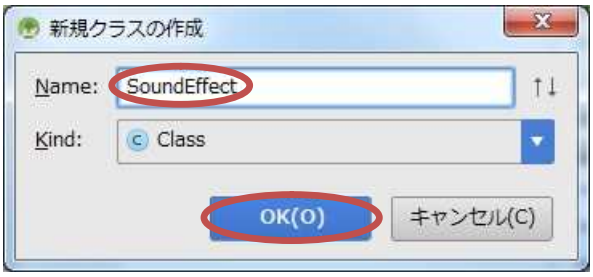


『Project』タブを開いて，これらの音データファイルが見えていることを確認する。



『Project』タブを開いた状態で，『java』→『jp.ac.cuc.b6x0004.prog_9th』と選択した上で，『ファイル』→『New』→『クラス』と選択する。

クラスの『Name:』を「SoundEffect」とする。

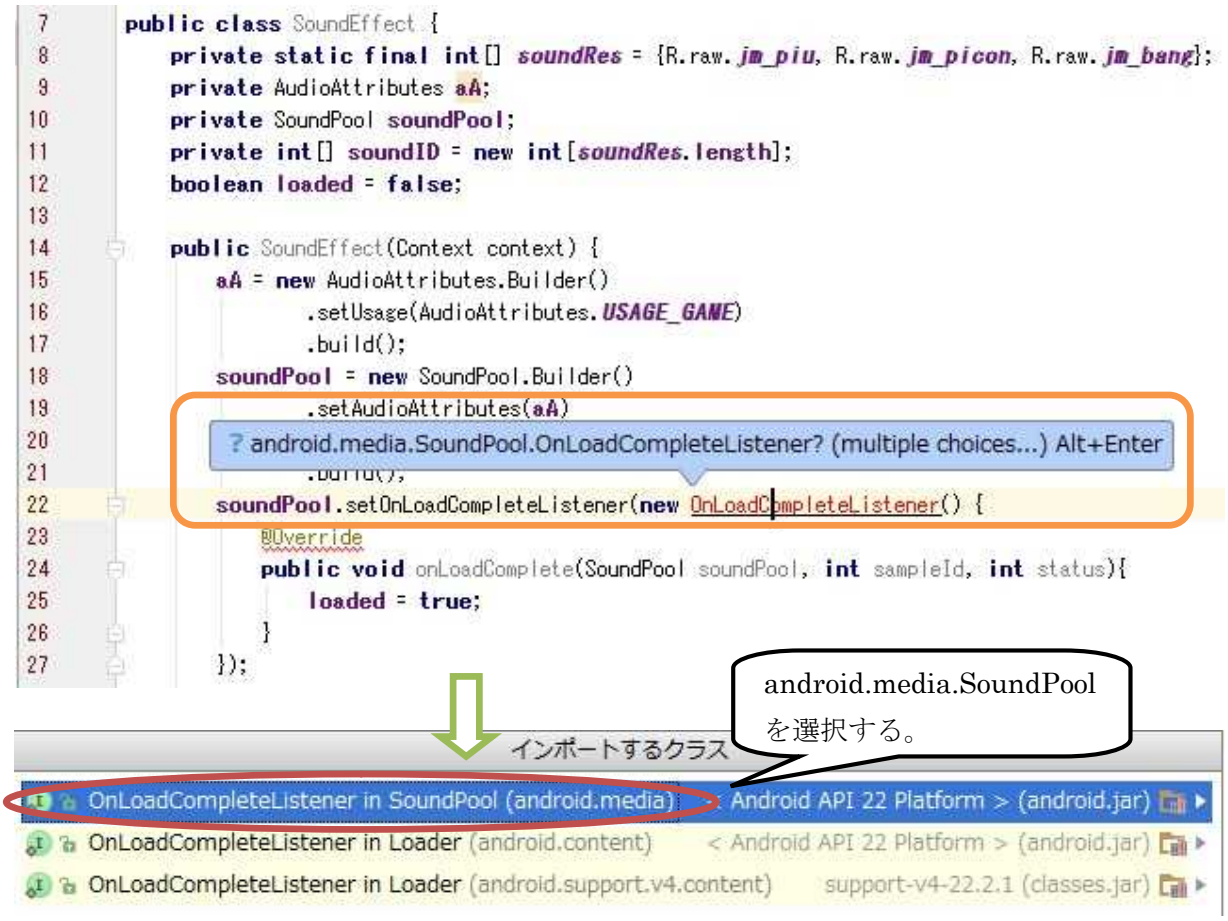


SoundEffect. java に次の内容を付け加える。(pp.2-3)

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

public SoundEffect(Context context) {
    aA = new AudioAttributes.Builder()
         .setUsage(AudioAttributes.USAGE_GAME)
         .build();
    soundPool = new SoundPool.Builder()
         .setAudioAttributes(aA)
         .setMaxStreams(1)
         .build();
    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);
    }
```

上記の内容を記述していく途中で，コンストラクタの中の soundPool に働きかけている setOnLoadCompleteListener メソッドの引数の OnLoadCompleteListener にカーソルを合わせると，何を選択するかという問いが現れるので，Alt キーを押しながら Enter キーを打つ。



activity_main9.xml × | SoundEffect.java × | MainActivity9.java ×

```
1   package jp.ac.cuc.b6x0004.prog_9th;
2
3   import android.content.Context;
4   import android.media.AudioAttributes;
5   import android.media.SoundPool;
6
7   public class SoundEffect {
8       private static final int[] soundRes = {R.raw.jm_piu, R.raw.jm_picon, R.raw.jm_bang};
9       private AudioAttributes aA;
10      private SoundPool soundPool;
11      private int[] soundID = new int[soundRes.length];
12      boolean loaded = false;
13
14      public SoundEffect(Context context) {
15          aA = new AudioAttributes.Builder()
16                  .setUsage(AudioAttributes.USAGE_GAME)
17                  .build();
18          soundPool = new SoundPool.Builder()
19                  .setAudioAttributes(aA)
20                  .setMaxStreams(1)
21                  .build();
22          soundPool.setOnLoadCompleteListener(new SoundPool.OnLoadCompleteListener() {
23              @Override
24              public void onLoadComplete(SoundPool soundPool, int sampleId, int status){
25                  loaded = true;
26              }
27          });
28
29          for (int i = 0; i < soundRes.length; i++)
30              soundID[i] = soundPool.load(context, soundRes[i], 1);
31      }
32
33      public void destroy() {
34          for (int i = 0; i < soundRes.length; i++)
35              soundPool.unload(soundRes[i]);
36
37          soundPool.release();
38      }
39
40      public void playSoundEffect(int i) {
41          int loop;
42
43          if (i == 1)
44              loop = 2;
45          else
46              loop = 0;
47
48          if (loaded)
49              soundPool.play(soundID[i], 1.0f, 1.0f, 1, loop, 1.0f);
50      }
51
52  }
```

==【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

aA・・・AudioAttributes クラスのインスタンス

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)

aA = new AudioAttributes.Builder()　・・・AudioAttributes オブジェクトの Builder クラス
.setUsage(AudioAttributes.USAGE_GAME)　・・・音データのタイプを記述
.build();　・・・音データについて記述された属性を統合して，オブジェクトを生成

soundPool = new SoundPool.Builder()　・・・SoundPool オブジェクトの Builder クラス
.setAudioAttributes(aA)　・・・音データの属性として何を使うかを指定
.setMaxStreams(1)　・・・同時に再生する音データの最大数（1 以上の整数）
.build();　・・・オブジェクトを生成

setOnLoadCompleteListener メソッド・・・OnLoadCompleteListener インタフェー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。

==========

＊ AudioAttributes.Builder についての詳細は，https://developer.android.com/reference/android/media/AudioAttributes.Builder.html を参照。
＊＊ SoundPool.Builder についての詳細は，https://developer.android.com/reference/android/media/SoundPool.Builder.html を参照。

『activity_main9.xml』のタブを開き，「Hello World!」と表示されているテキストビューをダブルクリックし，『…』ボタンを押して出てきた『Resources』では『New Resource』→『New string Value』を選択して，その値を変更する。

Resource name:　click　　Resource value:　ボタンをクリック

『Palette』の『Widgets』から『Button』をドラッグして配置する。ボタンをダブルクリックし，『…』ボタンを押して出てきた『Resources』では『New Resource』→『New string Value』を選択して，その値を変更する。

button　　Resource name:　Sbt1　　Resource value:　Piu!

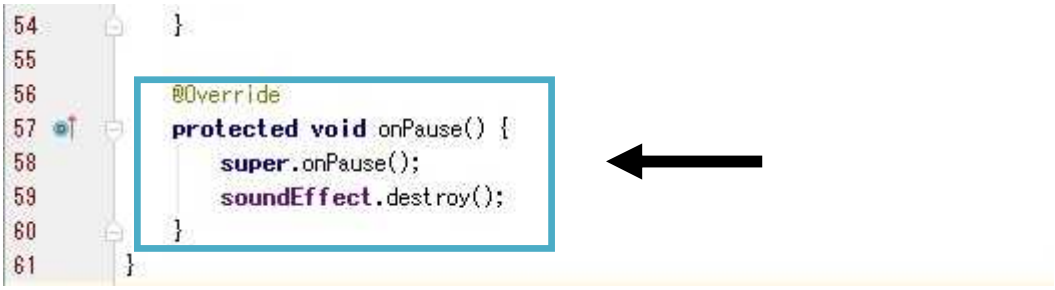
button2　　Resource name:　Sbt2　　Resource value:　Picon!×3

button3　　Resource name:　Sbt3　　Resource value:　Bang!

MainActivity9.java に色付きの枠の箇所を付け加える。

（pp.7-8 の図）

```
1   package jp.ac.cuc.b6x0004.prog_9th;
2
3   import android.content.Context;
4   import android.os.Bundle;
5   import android.support.v7.app.AppCompatActivity;
6   import android.view.View;
7   import android.widget.Button;
8
9   public class MainActivity9 extends AppCompatActivity {
10      public static final int SOUND_PIU = 0;
11      public static final int SOUND_PICON = 1;
12      public static final int SOUND_BANG = 2;
13      private SoundEffect soundEffect;
14
15      @Override
16      protected void onCreate(Bundle savedInstanceState) {
17          super.onCreate(savedInstanceState);
18          setContentView(R.layout.activity_main9);
19          Context context = getApplicationContext();
20          soundEffect = new SoundEffect(context);
21
22          Button btn = (Button)this.findViewById(R.id.button);
23          btn.setOnClickListener(
24                  new View.OnClickListener() {
25
26                      @Override
27                      public void onClick(View v) {
28                          soundEffect.playSoundEffect(SOUND_PIU);
29                      }
30                  }
31          );
32
33          Button btn2 = (Button)this.findViewById(R.id.button2);
34          btn2.setOnClickListener(
35                  new View.OnClickListener() {
36
37                      @Override
38                      public void onClick(View v) {
39                          soundEffect.playSoundEffect(SOUND_PICON);
40                      }
41                  }
42          );
43
44          Button btn3 = (Button)this.findViewById(R.id.button3);
45          btn3.setOnClickListener(
46                  new View.OnClickListener() {
47
48                      @Override
49                      public void onClick(View v) {
50                          soundEffect.playSoundEffect(SOUND_BANG);
51                      }
52                  }
53          );
```



```
54      }
55
56      @Override
57      protected void onPause() {
58          super.onPause();
59          soundEffect.destroy();
60      }
61  }
```

それぞれの色の枠内を次に示す。

```
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 btn = (Button)this.findViewById(R.id.button);
    btn.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　MainActivity9 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。

==========

『保存』のアイコンをクリックして，全てのファイルを上書き保存し，実行ボタンをクリックする。

提出物：

1）画面のレイアウト設定ファイル activity_main9. xml
2）SoundEffect のソースファイル SoundEffect. java
3）アクティビティのソースファイル MainActivity9. java