

Android ゲームプログラミング

第9回 サウンド

サウンド

- ・サウンドは、MediaPlayerクラス、AudioTrackクラス、SoundPoolクラスで扱うことができる
- ・MediaPlayerクラスは簡単に動画や音楽を再生できるが、再生までに少し時間がかかる
- ・AudioTrackクラスは再生までのスピードは早いですが、wav形式しか扱えない
- ・SoundPoolクラスは簡単にサウンドを再生でき、再生までのスピードも早い。効果音向き
- ・サウンドファイル、リソースフォルダに"raw"フォルダを作成し、保存する

MediaPlayerクラス

MediaPlayerクラスは、動画や音楽を簡単に再生することができます。反面、再生までにやや時間がかかります。ムービーやBGMを再生するのに向いています。

MediaPlayerクラスで音楽を再生するには、まずMediaPlayer.createメソッドで読み込み先のリソースまたはURLを指定します。その後、MediaPlayer.startメソッドで再生することができます。ループ再生を行いたい場合は、再生前にsetLoopingメソッドで設定しておきます。

```
// フィールド変数
private MediaPlayer    mMediaPlayer;
    :
// サウンドファイルの再生
mMediaPlayer = MediaPlayer.create(getContext(), R.raw.bgm1);
mMediaPlayer.setLooping(true);
mMediaPlayer.start();
```

リソースから読み込む場合は、リソースフォルダの"raw"というフォルダに保存する必要がありますが、デフォルトでは作成されません。このフォルダは、Eclipseなどから作成します。

再生の停止はMediaPlayer.stopメソッド、一時停止はMediaPlayer.pauseメソッドです。なお、stopメソッドを実行した場合は、prepareメソッドを実行しないと再度再生することができません。

```
// 停止
mMediaPlayer.stop();
try {
    mMediaPlayer.prepare();
} catch(Exception e) {
}
```

MediaPlayerに読み込んだデータが不用になった場合は、releaseメソッドで解放します。思わぬメモリリークを防ぐため、解放処理を記述した方が無難です。

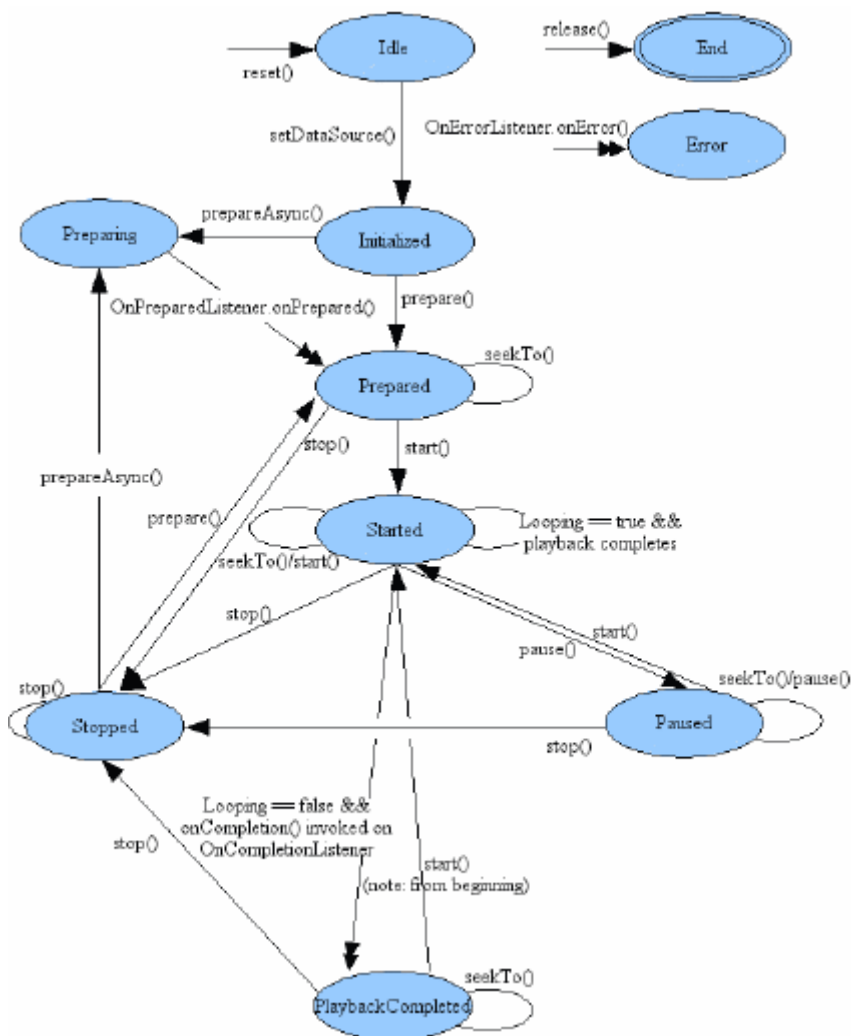
```
// MediaPlayerの解放
if(mMediaPlayer != null) {
    mMediaPlayer.stop();
    mMediaPlayer.release();
    mMediaPlayer = null;
}
```

なお、複数同時に再生することもできます。その分だけMediaPlayer型の変数が必要となります。変数は(メモリの許す限り)いくつでも宣言できますが、同時に再生できる数は機種により異なります。

MediaPlayerクラス対応形式 : wav, mp3, mp4, 3gp, ogg, midなど

MediaPlayerクラスの状態遷移

MediaPlayerクラスは、以下のように状態を持っており、実行できるメソッドは状態に依存します。状態に合わないメソッドを実行すると、例外"IllegalStateException"が投げられます。



AudioTrackクラス

AudioTrackクラスは、サウンドのデータをハードウェアに直接書き込みます。そのためMediaPlayerクラスより再生までの時間が早いのが特徴です。また、大きなサウンドデータも少しずつメモリに読み込みながら再生することもできます。しかし、wav形式しか扱えないため、mp3やogg形式などの圧縮されているファイルは、デコードするプログラムを作成する必要があります。また、正常に動作しない実機も多いようです。

AudioTrackクラス対応形式 : wavのみ

SoundPoolクラス

SoundPoolクラスは、データをデコードしてメモリに読み込んでおくため、再生時にほとんど遅延が生じません。読み込むと必ずメモリに配置されるため、大きなサウンドはメモリを圧迫する原因となったり、そもそも読み込めない場合もあります。そのため、短い時間の効果音に向いています。

SoundPoolクラスでサウンドを再生するには、まずSoundPoolクラスのコンストラクタでSoundPool型の変数を生成し、loadメソッドでリソースまたはファイルからサウンドを読み込みます。その後、playメソッドで再生することができます。再生回数の指定やループ再生も行えます。SoundPool型の変数1つで複数同時再生に対応しています。

```
// フィールド変数
private SoundPool mSoundPool;
private int[] mSoundID = new int[8]; // サウンドID
:
// SoundPoolの生成
mSoundPool = new SoundPool(mSoundID.length, AudioManager.STREAM_MUSIC, 0);

// サウンドの読み込み
mSoundID[0] = mSoundPool.load(getContext(), R.raw.se1, 1);
mSoundID[1] = mSoundPool.load(getContext(), R.raw.se2, 1);
mSoundID[2] = mSoundPool.load(getContext(), R.raw.se3, 1);
:
// サウンドの再生
mSoundPool.play(mSoundID[0], 1.0f, 1.0f, 0, 0, 1.0f);
```

SoundPool.loadメソッドは、SoundIDを返します。このIDは、loadされるたびに1, 2, 3...と増やされていきます。ところが、playメソッドに256以上の数値をIDに指定しても再生されません。

また、loadメソッドは読み込みが完了する前に制御が戻され、次の命令が実行されます。そのため、loadしてすぐにplayすると再生できない場合があります。この問題を解決するため、Android 2.2からは、load完了を通知できるようになっています。

読み込んだサウンドの解放はunloadメソッドで行います。不要になったサウンドは、unloadメソッドを呼び出してメモリを空けるようにします。

なお、SoundIDは、releaseメソッドを実行するまで、unloadしたとしても同じIDが使用されることはありません。SoundPoolを使い回す場合は、適宜releaseメソッドを実行します。

書式 SoundPool(int maxStreams, int streamType, int srcQuality)

maxStreams	SoundPoolが扱う最大のサウンド数。この数値 = 同時再生数ではありません。同時再生数はこの数にかかわらず、機種依存です。
streamType	また、大きな数を入力すると機種によっては挙動がおかしくなります
srcQuality	ストリームのタイプです。通常はAudioManager.STREAM_MUSICを指定します
	サンプリングレートの品質。通常は0を指定します

書式 SoundPool.load(Context context, int resId, int priority)

context	このアプリケーションのコンテキスト。getContext()で取得できます
resId	"R.raw."から始まるリソースのID。この定数は、Eclipseでは自動的に生成されます
priority	サウンドの優先度。現在未使用で、将来のためにあります。1を指定しておきます

書式 SoundPool.play(int soundID, float leftVolume, float rightVolume, int priority, int loop, float rate)

soundID	再生するサウンドのID。loadメソッドの戻り値です
leftVolume	左スピーカーの音量。0.0から1.0の間で指定。0.0は無音、1.0は最大音量
rightVolume	右スピーカーの音量。0.0から1.0の間で指定。0.0は無音、1.0は最大音量
priority	サウンド再生の優先度。0.0以上の数値で指定。0.0はもっとも低い優先度
loop	ループの回数。-1は無限、0は1回だけ再生、1は1回繰り返すので計2回再生
rate	再生速度。0.5から2.0の間。1.0は1倍速、0.5は0.5倍速、2.0は2倍速再生

SoundPoolクラスは、wavやmp3形式だと実機で挙動がおかしくなる場合があります。ogg形式であれば、ほとんどの機種で正常に動作するようです。

```
// サウンドの解放
if(mSoundID[0] != 0) {
    mSoundPool.stop (mSoundID[0]);
    mSoundPool.unload(mSoundID[0]);
    mSoundID[0] = 0;
}

:

// SoundPoolの解放
if(mSoundPool != null) {
    mSoundPool.release();
    mSoundPool = null;

    // SoundIDの初期化(必要な場合)
    for(int i = 0; i < mSoundID.length; i++)
        mSoundID[i] = 0;
}
```

SoundPoolクラス対応形式 : wav, mp3, mp4, 3gp, ogg, midなど

課 題

MediaPlayerクラスでMP3形式のサウンド、SoundPoolクラスで複数のogg形式の効果音を再生してみましょう。