

iアプリ Java ゲームプログラミング

第10回 音楽の再生

音楽ファイルの読み込みと再生

音楽ファイルを読み込み、再生するには、以下のクラスライブラリを使用します。

- AudioPresenter
- MediaSound

音楽ファイルは、MediaSoundで読み込みます。MediaSound.getSoundメソッドに音楽ファイルが保存されている場所とファイル名を指定すると、形式およびアクセス可能な場所かどうかのチェックを行います。問題がなければ、音楽ファイルを管理するためのMediaSound型の変数(オブジェクト)が返されます。次に、MediaSoundのuseメソッドを呼び出し、使用の宣言を行います。すると、実際に読み込まれ、内部形式に変換され、再生の準備が整います。

扱うことのできる形式は、.mid(標準MIDIファイル)と.mldです。指定できる場所は、リソース、URL、スクラッチパッド、byte配列ですが、大文字と小文字が区別されます。

音楽の再生は、AudioPresenterクラスで行います。useメソッドの呼び出しまで終わったMediaSoundオブジェクトをAudioPresenterのsetSoundメソッドに渡し、AudioPresenterのplayメソッドを呼び出せば、演奏が始まります。

- 音楽ファイル読み込みと再生のサンプルプログラム -

```
// 音楽ファイルを読み込むのと再生するのに必要な変数の宣言
AudioPresenter ap = AudioPresenter.getAudioPresenter();
MediaSound ms; // オーディオ再生オブジェクトの取得

// 場所の指定(リソース)
ms = MediaManager.getSound("resource:///ファイル名");

try {
    ms.use(); // 使用を宣言し、メモリに読み込む
    ap.setSound(ms); // AudioPresenterにセット
    ap.play(); // 再生開始
} catch (Exception e) {
    // 例外処理(上記の処理ができなかったときの処理をここに記述します)
}
```

音楽ファイルの解放

音楽ファイルの解放は、以下のような処理になります。

```
// 音楽ファイルの解放
try {
    ap.stop(); // 停止しておく(なくてもよい)

    ms.unuse(); // 使用終了の宣言
    ms.dispose(); // リソースの破棄
} catch (Exception e) {
}
ms = null;

// おまけのゴミ掃除(ガベージコレクション)
System.gc();
```

まず、MediaSoundのunuseメソッドを呼び出してリソースの使用終了を宣言します。最後に、disposeメソッドでMediaSoundを解放します。

音楽ファイルのイベント

AudioPresenterクラス、MediaSoundクラスともに、再生状態を調べることができません。再生が終了した場合は、そのまま放置されてしまいます。なんらかの工夫をしなければ、繰り返し再生できないことになってしまいます。

Star(DoJaもですが)では、音楽ファイルに何らかのイベントが起こると、それを通知させられる仕組みが用意されています。ただし、デフォルトでは通知すらされない状態となっているため、キャンバスクラスの宣言時から、イベント通知の実装宣言を行う必要があります。

- イベント通知の実装 -

1. MediaListenerの実装を宣言

```
// キャンバスクラスの定義時に、MediaListenerの実装も宣言する
public final class MyCanvas extends Canvas implements MediaListener {
```

キャンバスクラスの定義時に、MediaListenerインターフェースの実装を宣言します。これを行うとキャンバスクラスでイベントを受け取るためのmediaActionメソッドを定義できるようになります。

2. メディアリスナの設定

```
// メディアリスナ設定(一度だけ行います)
ap.setMediaListener((MediaListener)this);
```

AudioPresenterのsetMediaListenerメソッドに「implements MediaListener」を宣言したキャンバスオブジェクトを渡します。通常はキャンバスクラス内で行いますので、自分自身「this」となります。この設定により、オーディオ関連のイベントが起こるたびに下記のmediaActionメソッドが自動的に呼び出されるようになります。

3. メディアアクション処理(mediaActionメソッド)の定義

```
// --- メディアアクション処理(オーディオにイベントがあると呼び出されます)
public void mediaAction(MediaPresenter mp, int type, int param) {
    switch(type) {
        // 再生完了
        case AudioPresenter.AUDIO_COMPLETE:
            mp.play(); // 再生が終了したら始めから再生する
            break;

        // 再生中断
        case AudioPresenter.AUDIO_STOPPED:
            mp.play(); // 何らかの理由で再生が中断された場合も、再度再生する
            break;
    }
}
```

イベントが起きたときに呼び出されるmediaActionメソッドを定義します。このメソッドを作成しないと、エラーになります(MediaListenerがインターフェースのため)。

- MediaListener#mediaAction -

```
public void mediaAction(MediaPresenter mp, int type, int param)
mp      - イベントが起きたAudioPresenterもしくはVisualPresenterが渡されます
type    - イベントのタイプが渡されます
param  - イベントのパラメータが渡されます。ない場合は0が入ります
```

typeには、起きたイベントにより、下記の値が入ります(定義はAudioPresenterクラス)。

AUDIO_PLAYING	再生が開始された
AUDIO_COMPLETE	再生が終了した
AUDIO_PAUSED	再生が一時停止した
AUDIO_STOPPED	再生が停止された
AUDIO_RESTARTED	再生が再スタートした
AUDIO_LOOPED	ループが完了した

音楽ファイルを読み込み、再生しましょう。

(1)音楽ファイルを読み込み、再生してみましょう。

(2)音楽ファイルが繰り返し再生されるようにしましょう。

応用問題 ソフトキーで音量を変えられるようにしましょう。