

Javaルキーでも大丈夫!

# 勝手に作ろう!

# iアプリ

## 第 iアプリでiメロディーを奏でよう!

### 4 回

木寺祥友 + 株式会社エル・カミノ・リアル  
www.ecreal.co.jp

## 音楽を使う際のポイントは データ形式と機種依存

今月はiアプリで音楽を奏でる方法について解説する。これを使えばアクションやイベントに対応して音楽を使えるため、単なるウェブブラウジングでは実現できないアプリケーションならではの表現ができるはずだ。

実際にソースをどのように記述するかなどのくわしい説明は解説のなかで述べるとして、ここではiアプリで音楽を扱ううえでの注意点を挙げておこう。

まず1つ目は音楽のデータ形式だ。iアプリではメロディーというiアプリ専用の音楽データ形式しか使えない。そのためメロディーの作成は、下で解説している「PS-PLAYER」などの変換ソフトを利用する必要がある。

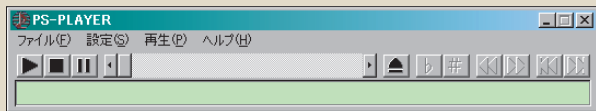
次に注意すべきは、DoJaはiメロディーを再生できないという点だ。先ほどiメロディーを用意しようとしたが、DoJaでiアプリを事前にテストする場合は、音楽データはMIDI形式でなくてはならない。それに合わせてソースも書き直して、コンパイルしておこう。

注意点の3つ目は機種依存だ。端末によっては指定したパートの音楽に対応していないため、思ったおりの音色が出ない可能性がある。実際に作成してから自分の端末などで確認してもらいたい。

以上のことを注意すれば、音楽を奏でること自体はさほど難しいことはない。解説を読んだうえでチャレンジしてみよう。



### PS-PLAYERでiメロディーに変換しよう



PS-PLAYER  
www.asahi-net.or.jp/~wj3a-fji/pbs/program/

ここでiメロディーを作成するうえで便利なフリーウェアソフト「PS-PLAYER」を紹介しよう。このソフトはNTTドコモやJ-フォンなど各社の携帯やPHS用の着メロデータを読み込んでパソコン上で再生できるオーディオプレイヤーで、MIDIから各着メロデータに変換できるコンバーター機能が付いている。これを使えばMIDIからiメロディーを簡単に作成できる。また、このソフトは単純に変換するだけでなく次のような機能が使えるのが便利だ。

- ① 開始位置と終了位置とを指定することで部分的な変換ができるため、ファイルサイズを自分でコントロールできる。
- ② 先頭に休符があるMIDIファイルでも、先頭休符をカットして変換できるため音の頭が無音にならない。

フリーなうえに185Kバイトと軽量で気軽に使える。さっそく「今月のオマケ!」にあるカップメンタイマーの音楽をカスタマイズするのに使ってみてはいかがだろうか。

# iメロディーを再生する プレイヤーを用意しよう

iメロディーを奏でるうえで主役となるのがAudioPresenterクラスとMediaSoundクラスだ。ここのソースは一見難しそうだが、それぞれのクラスの機能を具体的なものに置き換えて考えれば、それほど難しくはないだろう。

## AudioPresenterとMediaSoundのイメージ【41-45行目】

ここでは、iメロディーに関連するAudioPresenterクラスとMediaSoundクラスをTopFrmクラス全体の変数として定義している。これらのクラスの関係は、下のように音楽CDをCDプレイヤーの関係にたとえられる。

AudioPresenter ...> CDプレイヤー

MediaSound ...> 音楽CD

この関係を頭に入れたうえで以下の解説を読めば理解しやすいだろう。具体的なiメロディーを演奏させる手順は、TopFrm.javaの124行目から136行目までの部分になる。それぞれの手順を見ていこう。

## プレイヤーを準備する【126行目】

最初の手順では、AudioPresenterクラスのgetAudioPresenterメソッドを使ってAudioPresenter(のインスタンス)を準備している。これは、CDプレイヤーの電源を入れているようなものだ。

## 音源を用意する【130行目】

次の手順は、MediaManagerクラスのgetSoundメソッドを使ってMediaSoundクラス(のインスタンス)を準備している。これは、音楽CDをCDラックから取り出しているようなものだ。このMyIntroでは、iメロディーのデータをいったんスクラッチパッドにダウンロードしてから使っているため、getSoundの引数にスクラッチパッド内のiメロディーの位置を指定している。もちろんリソースファイル名やネット上にあるファイルを直接指定することもできる。

## 音源を使用可能にする【132行目】

次の手順はMediaSoundクラスのuseメソッドを使って音源を使用可能な状態にしている。音楽CDをケースから取り出すといった感じだ。余計な手順と感じるかもしれないが、iメロディーを使うには必要な手順だ。忘れないようにしよう。

## 演奏を開始する【134行目】

次の手順は、AudioPresenterクラスのsetSoundメソッドを使ってAudioPresenterにMediaSoundをセットしている。これはプレイヤーに音楽CDを入れているようなものだ。

## 演奏を開始する【136行目】

実際のCDプレイヤーでは再生ボタンを押すと音楽が流れるが、iアプリの再生ボタンを押す行為に当たるのがAudioPresenterのplayメソッドだ。これによってiメロディーが奏でられるのだ。

TopFrm.java

```
41 // 変数
42 private Button      btnOpn; // オープニングボタン
43 private Button      btnPro; // プロフィールボタン
44 private AudioPresenter audio; // AudioPresenter
45 private MediaSound  sound; // サウンドデータ
46
47 /**
48  * 初期化を行うコンストラクタ
49  */
50 public TopFrm()
51 {
52
```

(中略)

```
121 // サウンドの準備
122 try
123 {
124     // --- Sound を AudioPresenter に設定し、
再生できるようにする ---
125     // オーディオ制御を行えるようにする
126     audio = AudioPresenter.getAudioPresenter();
127     // メディアリスナを登録する
128     audio.setMediaListener( this );
129     // サウンドをリソースから読み込む
130     sound = MediaManager.getSound
( "scratchpad://0;pos=" + pMd );
131     // サウンドを使用可能にする
132     sound.use();
133     // サウンドをオーディオプレゼンターに設定する
134     audio.setSound( sound );
135     // サウンドの再生
136     audio.play();
137 }
138 catch( Exception e )
139 {
140     MyIntro.dlg( Dialog.DIALOG_ERROR, "エラー", "サウン
ドの設定に失敗しました。 " );
141 }
142
143 }
```

: 変数名    : コメント    : 外部ファイル名やカスタマイズ可能など

# iメロディーをループで再生させよう

前のページで説明した方法では、iメロディーの演奏が1回終了するとそのまま終了してしまう。もしiメロディーを繰り返して演奏させたい場合には、ループの仕掛けが必要だ。ここではメディアリスナーを使ってループさせる方法を解説しよう。

## メディアリスナー

メディアリスナーとは、前月号で紹介したイベントリスナーの1つでマルチメディア関係のイベントが発生したときに処理をする。ループ再生を行う具体的な方法としては、「iメロディーの再生が終了した」というイベントが起きた際に、再び再生を指示すればよい。

TopFrmにMediaListener機能を持たせる【16行目】

クラス宣言の部分のimplements以降にMediaListenerを付け加える。これで、このTopFrmクラスのmediaActionメソッドでマルチメディア関係のイベントを受け取る準備ができたことになる。

メディアリスナーを登録する  
【128行目(前ページ)】

この部分でaudioから発生したイベントをこのクラス(TopFrmクラス)で受け取るように指定している。これで、「iメロディーの再生が終了した」というイベントがこのTopFrmクラスのmediaActionメソッドに知らされるようになる。

演奏が終了したら再び再生する  
【212-222行目】

iメロディーの演奏が終了すると、mediaActionメソッドにイベントが通知される。「演奏の終了」を表すイベントには、引数のtypeにAudioPresenter.AUDIO\_COMPLETEが設定されているため、if文を使って、演奏完了の場合に再びplayメソッドで再生を開始している。これで、ループ再生を実現しているのだ。

：変数名   ：コメント  
：外部ファイル名やカスタマイズ可能なところ

```

TopFrm.java
13 /**
14  * トップパネル
15  */
16 class TopFrm extends Panel implements SoftKeyListener,
    ComponentListener, MediaListener
17 {
    (中略)

145 /**
146  * ソフトキーが押された時に呼ばれる
147  */
148 public void softKeyPressed( int softKey )
149 {
    (中略)

156 // ソフトキー-2
157 if ( Frame.SOFT_KEY_2 == softKey )
158 {
159 // 更新
160 // 背景を赤く
161 setBackground( Graphics.getColorOfName( Graphics.RED ) );
162 // サウンドの破棄
163 audio.stop();
164 sound.dispose();
165 // データの更新
166 loadData();
167 readScratch();
168 // パネル準備
169 MyIntro.iniProF();
170 // サウンドの準備
171 try
172 {
173     sound = MediaManager.getSound( "scratchpad:///0;pos=" + pMd );
174     sound.use();
175     audio.setSound( sound );
176     audio.play();
177 }
178 catch( Exception e )
179 {
180     MyIntro.dlg( Dialog.DIALOG_ERROR, "エラー", "サウンドの設定に
        失敗しました。" );
181 }
182 // 背景を戻す
183 setBackground( Graphics.getColorOfRGB( BG_P[ 0 ], BG_P[ 1 ],
        BG_P[ 2 ] ) );
184 }
185 }
    (中略)

212 /**
213  * メディアプレゼンタから通知を受けたときに呼ばれる
214  */
215 public void mediaAction( MediaPresenter source, int type,
    int param )
216 {
217 // 曲をループ再生させるために、play()を呼ぶ
218 if( type == AudioPresenter.AUDIO_COMPLETE )
219 {
220     audio.play();
221 }
222 }
    
```

# iメロディーを更新するまえに 後始末をしよう

新しいiCDを聞こうとすると、それ以前にプレイヤーに入れていたCDを取り出す必要がある。これと同じでiメロディーでも新しい音楽をダウンロードしようとする場合には、その前に再生していた音楽を片付ける必要がある。曲を変えるときには忘れずに行っておこう。

## 演奏の停止【163行目】

ここではCDプレイヤーの停止ボタンを押しているのと同じだ。AudioPresenterクラスのstopメソッドを使って演奏を停止させている。

## 音源の破棄【164行目】

MediaSoundクラスのdisposeメソッドを使って、音源を破棄している。音楽CDをプレイヤーから取り出しているのをイメージすればよい。



## 今月のオマケ!

3分で完成するオリジナルiアプリ

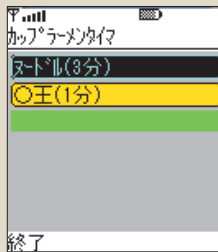
# 「オリジナルメロディーの タイマーをつくらう」

ここでは、ちょっとした変更だけで作れる自分だけのiアプリを紹介していく。今月は自分で用意した音楽を流せるタイマー「カップメンタイマー」だ。

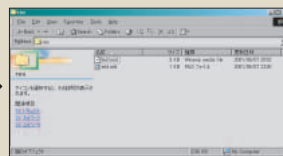
### 1 ダウンロードしよう

まずはカップメンタイマーをダウンロードしよう。インターネットマガジンのサイト内にあるiアプリのコーナーからダウンロードしよう。

[internet.impress.co.jp/iappli/](http://internet.impress.co.jp/iappli/)

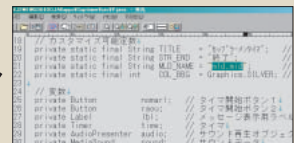


### 2 MIDIデータを用意しよう



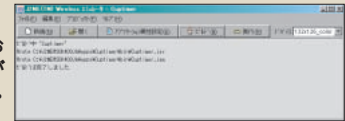
それでは実際にタイマーで鳴らす音楽をMIDIで用意して、RESフォルダーに入れておこう。この際MIDIデータを入れるのは、本文中でも触れたように、DoJaがメロディーに対応していないので、MIDIでないとはテストできないためだ。ちなみにMIDIのサイズはメロディーにしたとき8084バイト以内に収まる程度にしておこう。

### 3 ソースを書き換えよう



それではsrcフォルダーにあるP.javaのソースを変更しよう。21行目にある「mid.mid」というファイル名を先ほどRESフォルダーに入れたMIDIデータに書き換えればよい。この際、拡張子もMIDIデータの.midにするのを忘れないように。

### 4 ビルドしよう



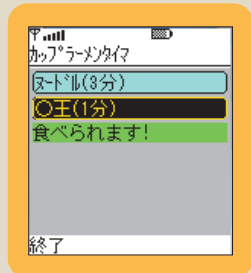
ソースを書き換えたらビルドしなそう。その際、ファイルの属性が「読み取り専用」だとエラーになる。ビルドする前にはずしておこう。

### 5 DoJaでテストをしよう

それでは実際にDoJaで実行してみよう。用意したMIDIデータが再生されたら成功だ。



### 6 完成!



成功したら、MIDIデータをiメロディーデータにコンパットして、先ほどのソースもそれに合わせて書き直そう。これでホームページサーバーを経由して携帯電話にダウンロードすればよい。



もっと知りたいキミにはコレ!

『今すぐできるiアプリプログラミング』

小社発行 本体価格2,400円

来月なんて待ってられない、もっとくわしくいろいろと知りたい! と思っている君にはコレがオススメ。本連載の木寺氏が執筆しているため、ここで取り上げているiアプリについて、もっとくわしく解説している。一緒に読んでいけば、理解が深まるぞ。



## [インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

**株式会社インプレスR&D**

All-in-One INTERNET magazine 編集部

[im-info@impress.co.jp](mailto:im-info@impress.co.jp)