

# iアプリ Java ゲームプログラミング

## 第7回 画像の読み込み

### 画像の読み込み

画像を読み込むには、以下のクラスライブラリを使用します。

- MediaManager
- MediaImage
- Image

画像は、MediaManagerをとおして取得します。MediaManager.getImageメソッドに画像が保存されている場所とファイル名を指定すると、形式のチェック、アクセス可能な場所かどうかのチェックを行います。誤りがなければ、画像を管理するためのMediaImage型の変数が返されます。

扱うことのできる形式は、GIF、透過GIF、JPEG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るので、注意が必要です。

MediaImageは、画像を管理します。MediaImageが指定された場所から画像を読み込み、各機種に合わせたデータに変換します。そのほか、画像の幅、高さ、イメージデータの取得、読み込まれた画像をメモリから解放する機能を提供します。

画像を描画するには、Image型の変数をGraphicsのdrawImageメソッドに渡します。Image型の変数は、MediaImageのgetImageメソッドで取得できます。

#### - 画像読み込みのサンプルプログラム -

```
// 画像を読み込むのと描画するのに必要な変数の宣言
MediaImage mi;
Image img;

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

// 画像の使用を宣言し、メモリに読み込む
try {
    mi.use();
} catch(Exception e) {
    // 例外処理(読み込めなかったときの処理をここに記述します)
}

// イメージの取得
img = mi.getImage();
```

useメソッドは、try-catch文で囲まないとエラーになります。try-catch文は、重大なエラーが発生したとき、どのようにするかを記述するためのものです。

画像を描画するには、GraphicsのdrawImageメソッドを呼び出します。このメソッドには、Image型の変数と描画座標(イメージの左上が基準)を渡します。

```
// 画像の描画(画面の(120, 80)に描画)
grp.drawImage(img, 120, 80);
```

## 画像の解放

携帯電話は容量がかなり少ないので、たくさんの画像をメモリに読み込むことはできません。不要になった画像を解放し、新しい画像を読み込むなどの工夫が必要になります。

画像の解放は、以下のような処理になります。

```
// 画像使用の終了を宣言(全機種共通)
try {
    img.dispose();
    img = null;

    mi.unuse();
    mi.dispose();
    mi = null;
} catch(Exception e) {
}

// ゴミ掃除(ガベージコレクション)
System.gc();
```

Image型の変数を取得したので、これをdisposeメソッドにより解放します。次に、MediaImageのunuseメソッドを呼び出してメモリから画像を解放します。最後に、disposeメソッドでMediaImageを解放します。最後のdisposeメソッドを呼ばない限り、メモリから解放されない機種もあります(プログラムを終了した場合は解放されます)。

Nシリーズなど一部の機種では、ImageをdisposeしただけでMediaImageもdisposeされてしまうという不具合があるので、上記のようにtry-catchで囲んでおきます。

## 課題

画像を読み込み、描画しましょう。

(1) 240×240の画像を用意しましょう。保存形式は、JPGまたはGIFにします。これを背景とします。

(2) 作成した画像を、プロジェクトのresフォルダに移動しましょう。

(3) 以下のように、MediaImage型とImage型の変数をキャンバスで宣言しましょう。

```
MediaImage mi;
Image img;
```

(4) 画像を読み込みます。読み込みは、何度も行う必要がないので、initメソッドに記述します。initメソッドを以下のように変更しましょう。

```
// --- 初期化
void init() {
    // 場所の指定(リソース)
    mi = MediaManager.getImage("resource:///filename.jpg");

    // 画像の使用を宣言し、メモリに読み込む
    try {
        mi.use();
    } catch(Exception e) {
        // 例外処理(読み込めなかったときの処理をここに記述します)
    }

    // イメージの取得
    img = mi.getImage();
}
```

(5)読み込んだ画像を描画します。描画は、drawメソッドで行います。drawメソッドを以下のように変更しましょう。なお、「?」は各自考えてください。

```
// --- 描画
void draw() {
    grp.lock();                // 描画開始
    grp.????????(img, 0, 0);
    grp.unlock(true);        // 描画終了
}
```

(6)透過色を設定したキャラクターを読み込み、背景の上に描画しましょう。