

- サウンド (ファイル名 : sample.mp3)

Androidで利用できるサウンドのファイル形式は「MP3」です。

3-7-3 ソースコード

■ MediaPlayerEx クラス

MediaPlayerEx クラスは、プログラムの本体となるクラスです。

リスト3-7-1 MediaPlayerEx.java

```
package net.npaka.mediaplayerex;
import android.app.Activity;
import android.os.Bundle;
import android.view.Window;

//サウンドの再生
public class MediaPlayerEx extends Activity {

    //アプリの初期化
    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(new MediaPlayerView(this));
    }

    //アプリの停止
    @Override
    public void onPause() {
        super.onPause();
        //アプリの強制終了 (4)
        System.exit(0);
    }
}
```

■ MediaPlayerView クラス

MediaPlayerView クラスは、ビューとなるクラスです。

リスト3-7-2 MediaPlayerView.java

```
package net.npaka.mediaplayerex;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
```

MediaPlayerクラス

void start()

機能：サウンドの再生開始

(3) サウンドの停止

サウンドの停止を行うには、MediaPlayerクラスのpause()メソッドを使います。

MediaPlayerクラス

void pause()

機能：サウンドの再生停止

(4) アプリの強制終了

Androidアプリのライフサイクルでは、BACKボタンなどでアプリが見えなくなってもアプリは終了しません。メモリが足りなくなった時に終了します。今回は強制終了させたいので、onPause()メソッドをオーバーライドして、Systemクラスのexit()メソッドを呼んでアプリを強制終了させます。

Systemクラス

static void exit(int code)

機能：アプリの強制終了

引数：*code* リターンコード

リターンコードは、正常終了時は通常0を戻します。

7-1

カメラの制御

7-1-1 カメラの制御を使用したプログラムの構成

カメラを制御するプログラムを作ります。ここで紹介するサンプルプログラムでは画面いっぱいにリアルタイムのカメラ映像を表示します。そして画面にタッチすることにより、スクリーンショットをSDカードに保存します。保存するファイル名は「test.jpg」で、DDMSのFile Explorerでは「/sdcard/test.jpg」で閲覧できます。

サンプルプログラムは、次の2つのクラスで構成されています。プロジェクトは、プロジェクト名「CameraEx」で作成してください。

- CameraEx クラス (CameraEx.java)
- CameraView クラス (CameraView.java)



横画像に差し替え

図 7-1-1 作成するサンプルプログラム

7-1-2 ソースコード

■ CameraEx クラス

CameraEx クラスは、プログラムの本体となるクラスです。

リスト7-1-1 CameraEx.java

```
package net.npaka.cameraex;
import android.app.Activity;
import android.os.Bundle;
import android.view.Window;

//カメラの制御
public class CameraEx extends Activity {
    //アプリの初期化
    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(new CameraView(this));
    }

    //アプリの強制停止
    @Override
    public void onPause() {
        super.onPause();
        //アプリの終了
        System.exit(0);
    }
}
```

■ CameraView クラス

CameraView クラスは、ビューなるクラスです。

リスト7-1-2 CameraView.java

```
package net.npaka.cameraex;
import android.content.Context;
import android.hardware.Camera;
import android.view.MotionEvent;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import java.io.FileOutputStream;

//カメラの制御
public class CameraView extends SurfaceView
    implements SurfaceHolder.Callback {
    private SurfaceHolder holder;//ホルダー
```

FileOutputStreamクラス

FileOutputStream(String fileName)

機能：FileOutputStreamクラスのコンストラクタ

引数：*fileName* ファイル名

SDカードに保存するには、パスの先頭を「/sdcard/」で始めます。write()メソッドでバイトデータを書き込み、close()メソッドでクローズします。

7-1-4 AndroidManifest.xmlの編集

今回はカメラ機能を使うので、AndroidManifest.xmlの「User Permission」の「android.permission.CAMERA」を追加します。追加の方法は、「4-5 ビデオビュー」のAndroidManifest.xmlの編集を参照してください。

また、今回は縦画面で表示していますが、横画面で表示したい場合は、AndroidManifest.xmlのactivityタグに「android:screenOrientation="landscape"」を追加します。

リスト7-1-3 AndroidManifest.xmlの追加

```
<activity android:name=".CameraEx"
          android:label="@string/app_name"
          android:screenOrientation="landscape">
```

さらに、アンテナや電池の情報も消して全画面表示にしたい場合は、Activityクラスを継承したクラス内で、以下のメソッドを呼んでください。

```
getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

LocationListenerインタフェース

```
void onStatusChanged(String provider, int status, Bundle extras)
```

機能：位置情報状態変更を通知する時に呼ばれる

引数：*provider* プロバイダー

status 状態

extras パラメータ

ここでは、位置情報変更時にマップビューに位置情報を指定するので、onLocationChanged()メソッドのみを利用します。他のメソッドは空とします。

(4) 緯度と経度の取得

緯度と経度を取得するには、LocationクラスのgetLatitude()メソッドと、getLongitude()メソッドを使います。

Locationクラス

```
double getLatitude()
```

機能：緯度の取得

戻り値：緯度

Locationクラス

```
double getLongitude()
```

機能：経度の取得

戻り値：経度

取得した緯度と経度を元にGeoPointオブジェクトを生成します。GeoPointオブジェクトの引数には、int型の値「緯度×1E6」と「経度×1E6」を指定します。

7-2-4 AndroidManifest.xmlの編集

このプログラムでは、通信機能を使うので「android.permission.INTERNET」を追加してください。さらにマップビューを使うので、「Uses Library」として「com.google.android.maps」を追加します。詳しくは「4-7 マップビュー」を参照してください。

加えて、位置情報機能を使うので「android.permission.ACCESS_FILE_LOCATION」、ネットワークによる位置情報取得機能を使うので「android.permission.ACCESS_COARSE_LOCATION」、エミュレータでの位置情報機能を使うので「android.permission.ACCESS_MOCK_LOCATION」を追加します。詳しくは「4-5 ビデオビュー」を参照してください。