Android Media (Audio & Video)

Sang Shin
Michèle Garoche
www.javapassion.com
“Learn with Passion!”

Disclaimer

- Portions of this presentation are modifications based on work created and shared by the Android Open Source Project
  - http://code.google.com/policies.html
- They are used according to terms described in the Creative Commons 2.5 Attribution License
  - http://creativecommons.org/licenses/by/2.5/

Topics

- Audio/Video support in Android platform
- Audio and Video playback
- Playing JET content
- Supported media formats
- Capturing audio

Audio and Video Support in Android Platform

Audio & Video Support in Android

- The Android platform offers built-in encoding/decoding for a variety of common media types, so that you can easily integrate audio, video, and images into your applications.
- Accessing the platform's media capabilities is fairly straightforward — you do so using the same intents and activities mechanism that the rest of Android uses.

Audio and Video Playback

Audio and Video Playback

- Media file can be obtained from multiple sources
  - /res/raw directory
  - File system
  - Network stream
- You can play back the audio data only to the standard output device; currently, that is the mobile device speaker or Bluetooth headset.

Play from a Raw Resource

- Play back media (notably sound) within your own applications.
- Steps to take
  - Put the sound (or other media resource) file into the */res/raw* folder of your project, where the Eclipse plugin (or aapt) will find it and make it into a resource that can be referenced from your R class
  - Create an instance of *MediaPlayer*, referencing that resource using *MediaPlayer.create*, and then call start() on the instance:

```
MediaPlayer mp = new MediaPlayer();
MediaPlayer mp = MediaPlayer.create(context,
                                    R.raw.sound_file_1);

mp.start();
```

Playing from a File or Stream

- You can play media files from the filesystem or a web URL
- Steps to take
  - Create an instance of the MediaPlayer
  - Call *setDataSource()* with a String containing the path (local filesystem or URL) to the file you want to play
  - Call *prepare()* then *start()* on the instance:

```
MediaPlayer mp = new MediaPlayer();
mp.setDataSource(PATH_TO_FILE);
mp.prepare();
mp.start();
```

Playing JET Content

What is JET?

- JET is an interactive music player for small embedded devices, including the Google Android platform.
- It allows applications to include interactive music soundtracks, in MIDI format, that respond in real-time to game play events and user interaction.

Playing JET Content

- The Android platform includes a JET engine that lets you add interactive playback of JET audio content in your applications.
- You can create JET content for interactive playback using the JetCreator authoring application that ships with the SDK.
- To play and manage JET content from your application, use the JetPlayer class.

Playing JET Content

- A example of how to set up JET playback from a .jet file stored on the SD card:

```
JetPlayer myJet = JetPlayer.getJetPlayer();
myJet.loadJetFile("/sdcard/level1.jet");
byte segmentId = 0;

// queue segment 5, repeat once, use General MIDI,
// transpose by -1 octave
myJet.queueJetSegment(5, -1, 1, -1, 0, segmentId++);
// queue segment 2
myJet.queueJetSegment(2, -1, 0, 0, 0, segmentId++);

myJet.play();
```

Android Supported Media Formats

Core Media Formats

- Audio
  - AAC LC/LTP, HE-AACv1 (AAC+), HE-AACv2 (enhanced AAC+), AMR-NB, AMR-WB, MP3, MIDI, Ogg Vorbis, PCM/WAVE
- Video
  - H.263, H.264 AVC, MPEG-4 SP

Audio Capture

Audio Capture

- The Android platform lets you record audio and video, where supported by the mobile device hardware.
- To record audio or video, use the *MediaRecorder* class.

Audio Capture Example

```
recorder = new MediaRecorder();
ContentValues values = new ContentValues(3);

values.put(MediaStore.MediaColumns.TITLE, SOME_NAME_HERE);
values.put(MediaStore.MediaColumns.TIMESTAMP, System.currentTimeMillis());
values.put(MediaStore.MediaColumns.MIME_TYPE, recorder.getMimeContentType());

ContentResolver contentResolver = new ContentResolver();

Uri base = MediaStore.Audio.INTERNAL_CONTENT_URI;
Uri newUri = contentResolver.insert(base, values);

if (newUri == null) {
    // need to handle exception here - we were not able to create a new
    // content entry
}

String path = contentResolver.getDataFilePath(newUri);

// could use setPreviewDisplay() to display a preview to suitable View here

recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
recorder.setOutputFile(path);

recorder.prepare();
recorder.start();
```

Thank you!

Check JavaPassion.com Codecamps!
http://www.javapassion.com/codecamps
"Learn with Passion!"