

Phát triển ứng dụng Smartphone

Tài liệu lưu hành nội bộ

Đây là tài liệu tham khảo sử dụng trong môn học Lập trình ứng dụng Smartphone – Android được tổng hợp, biên soạn từ nhiều nguồn bởi các thành viên của Nhóm nghiên cứu và ứng dụng công nghệ A106-Đại học Hoa Sen.



Phát triển ứng dụng Smartphone – Android

Phát triển ứng dụng Smartphone

Phần 07: Media

Lê Đức Huy

Email: leduchuy89vn@gmail.com



Mục lục

1	Chơi tệp tin âm thanh	3
1.1	Sử dụng phương thức setDataSource()	7
1.2	Chơi tệp tin âm thanh đặt trong thẻ nhớ SD	8
2	Chơi tệp tin Video	12
2.1	Chơi tệp tin video từ thẻ SD	13
3	Một số lưu ý khi làm việc với MediaPlayer và ViewView	13
4	Ghi âm thanh	14
5	Ghi video.....	18



Phát triển ứng dụng Smartphone – Android

Android hỗ trợ hàm chơi audio và video trong gói android.media. Phần chính của gói này là lớp android.media.MediaPlayer. Lớp này được sử dụng trong phần nghe audio và video, có thể đọc được nội dung từ các nguồn sau: web, tệp tin media đính kèm trong tệp .apk file, thẻ SD. MediaPlayer có thể đọc được các tệp có phần mở rộng sau: 3GPP (.3gp), MP3 (.mp3), MIDI (.mid and others), PCM/WAVE (.wav), and MPEG-4(.mp4).

Tạo mới một project với các thông số sau:

Project name: AudioExample

Build target: Android 2.3.3.

Application name: Audio Example

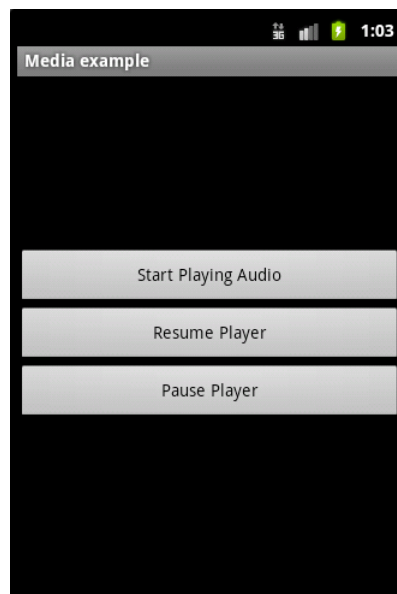
Package name: niit.android

Create Activity: main

1 Chơi tệp tin âm thanh

Đầu tiên, ta xây dựng ứng dụng đơn giản để chơi tệp tin MP3 từ Web. Sau đó ta sẽ sử dụng hàm `setDataSource()` để chạy nội dung từ file .apk hoặc từ thẻ nhớ.

Xây dựng một Activity dùng để chơi một tệp tin âm thanh với giao diện như sau:



Tham khảo nội dung tệp tin xml đánh dấu giao diện của activity dùng chơi nhạc ở trên:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
```



Phát triển ứng dụng Smartphone – Android

```
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center">
        <Button
            android:id="@+id/btnStartPlayer"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Start Playing Audio" />
        <Button
            android:id="@+id/btnResumePlayer"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Resume Player" />
        <Button
            android:id="@+id/btnPausePlayer"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Pause Player" />
    </LinearLayout>
```

Dùng một `AudioExampleActivity` với nội dung như sau:

```
package niit.android;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class AudioExampleActivity extends Activity {
    static final String AUDIO_PATH =
    "http://www.androidbook.com/akc/filestorage/android/documentfile
    s/3389/play.mp3";

    private MediaPlayer mediaPlayer;

    private int playbackPosition = 0;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.audio_example);
        Button btnStartPlayer =
        (Button) findViewById(R.id.btnStartPlayer);
        Button btnPausePlayer =
        (Button) findViewById(R.id.btnPausePlayer);
```




Phát triển ứng dụng Smartphone – Android

```
Button btnResumePlayer =
(Button)findViewById(R.id.btnResumePlayer);
btnStartPlayer.setOnClickListener(new
OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            playAudio(AUDIO_PATH);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
btnPausePlayer.setOnClickListener(new
OnClickListener() {
    @Override
    public void onClick(View view) {
        if (mediaPlayer != null) {
            playbackPosition =
mediaPlayer.getCurrentPosition();
            mediaPlayer.pause();
        }
    }
});
btnResumePlayer.setOnClickListener(new
OnClickListener() {
    @Override
    public void onClick(View view) {
        if (mediaPlayer != null &&
!mediaPlayer.isPlaying()) {
            mediaPlayer.seekTo(playbackPosition);
            mediaPlayer.start();
        }
    }
});
}

private void playAudio(String url) throws Exception {
    killMediaPlayer();
    mediaPlayer = new MediaPlayer();
    mediaPlayer.setDataSource(url);
    mediaPlayer.prepare();
    mediaPlayer.start();
}

private void killMediaPlayer() {
    if (mediaPlayer != null) {
```



Phát triển ứng dụng Smartphone – Android

```
        try {
            mediaPlayer.release();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        killMediaPlayer();
    }
}
```

Trong trường hợp này ta chạy một tệp Mp3 file từ địa chỉ web. Vì vậy ta cần thêm khai báo sau

```
<uses-permission android:name="android.permission.INTERNET" />
```

vào file manifest. Bạn có thể thấy trong hàm onCreate(), ta đã setClickListener cho cả 03 button.

Khi người dùng nhấn vào button **btnStartPlayer**, phương thức playAudio() sẽ được gọi. Trong phương thức playAudio(), một thể hiện của MediaPlayer được tạo ra, dữ liệu của player được gán là một đường dẫn đến một tệp tin MP3. MediaPlayer bắt đầu chơi tệp tin.

Khi người dùng nhấn vào button **bntPausePlayer**, ta lấy được vị trí hiện tại của player bằng hàm getCurrentPosition() và gán vào biến playbackPosition đã định nghĩa ở trên. Sau đó ta dừng việc chơi nhạc lại bằng phương thức pause().

Khi người dùng nhấn vào button **btnResumePlayer**, phương thức seekTo() được gọi với parameter là playbackPosition được truyền vào, khi ta gọi hàm start() mediaPlayer sẽ bắt đầu chơi tại vị trí playbackPosition thay vì từ đầu.

Lớp MediaPlayer còn chứa hàm stop(). Chú ý rằng nếu bạn stop player bằng hàm này. Bạn cần gọi hàm prepare() trước khi gọi start() một lần nữa. nhưng nếu bạn gọi pause(), bạn có thể gọi start ngay mà không cần gọi prepare().

Ngoài ra, hãy chắc chắn là bạn sẽ gọi hàm release() 1 lần khi bạn đã sử dụng xong đối tượng MediaPlayer. Trong ví dụ trên, ta đã làm việc này trong hàm killMediaPlayer().

Trong phần trên ta đã chạy một tệp tin audio từ web. Lớp MediaPlayer còn hỗ trợ chạy media từ file .apk. Để thực hiện việc này, ta thêm một thư mục tên raw vào trong thư mục /res nếu project của bạn chưa có. Sau đó, ta copy một tệp tin Mp3 vào /res/raw với tên là music_file.mp3.

Bổ sung thêm phương thức sau :

```
private void playLocalAudio() throws Exception
{
```



Phát triển ứng dụng Smartphone – Android

```
        mediaPlayer = MediaPlayer.create(this,
R.raw.music_file);
        mediaPlayer.start();
    }
```

Ở đây ta gọi làm static MediaPlayer.create() để tạo một thể hiện của MediaPlayer cho resource được truyền vào và sau đó gán nó cho thể hiện mediaPlayer của chúng ta và cuối cùng là start() nó.

1.1 Sử dụng phương thức setDataSource()

Ở trên chúng ta gọi hàm create để load file audio từ một raw resource. Với cách này, ta không cần phải gọi setDataSource(). Nhưng trong trường hợp thể hiện MediaPlayer của ta sử dụng constructor mặc định hoặc nội dung media không thể truy cập được thông qua resource ID hoặc URL như ở trên thì ta sẽ sử dụng phương thức setDataSource().

Phương thức setDataSource đã được overload thành nhiều version vì vậy bạn có thể sử dụng để tùy chỉnh datasource cho những gì bạn cần.

Tiến hành bổ sung thêm phương thức sau:

```
private void playLocalAudio_UsingDescriptor() throws Exception {
    AssetFileDescriptor fileDesc =
getResources().openRawResourceFd(R.raw.music_file);
    if (fileDesc != null) {
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setDataSource(fileDesc.getFileDescriptor(),
fileDesc.getStartOffset(), fileDesc.getLength());
        fileDesc.close();
        mediaPlayer.prepare();
        mediaPlayer.start();
    }
}
```

Phương thức getResources() sẽ trả về đối tượng kiểu Resources, đối tượng này sẽ được dùng truy xuất đến các tài nguyên đính kèm trong tập tin cài đặt ứng dụng (Hình ảnh, âm thanh...). Sau đó sử dụng phương thức openRawResourceFd() để lấy về một đối tượng kiểu AssetFileDescriptor, đây là đối tượng chứa các thông tin mô tả về tập tin audio chứa trong thư mục res/raw. Cuối cùng dùng phương thức setDataSource() với tham số truyền vào là đối tượng AssetFileDescriptor cùng với các thông số điểm bắt đầu chơi tập tin âm thanh là đầu tệp và điểm kết thúc là cuối tệp. Nếu bạn muốn chơi toàn bộ tệp tin thì có thể sử dụng một phương thức đơn giản hơn là setDataSource(FileDescriptor desc).

Sử dụng một trong các phương thức overload lại phương thức setDataSource() với một tham số truyền vào là FileDescriptor để chơi các tệp tin media đặt trong thư mục data của ứng dụng. Vì lý do bảo mật nên MediaPlayer không được phép truy xuất thư mục data của ứng dụng. Tuy nhiên, ứng dụng của bạn có thể mở một tệp tin và sử dụng nó trong phương thức setDataSource() thông qua đối tượng FileDescriptor. Giả sử bạn có một số tệp tin và thư mục chứa tệp tin media đặt tại địa chỉ



Phát triển ứng dụng Smartphone – Android

data/data/APP_PACKAGE_NAME/, bạn có thể gọi một số phương thức từ Activity đến lấy địa chỉ thư mục thay vì dùng địa chỉ cố định bằng code java.

VD: Dùng phương thức `getFilesDir()` để lấy đường dẫn đến tệp thư mục `data/data/APP_PACKAGE_NAME`, phương thức `getCacheDir()` để lấy thư mục chứa các tệp tin cache của ứng dụng.

Bằng cách này thì ứng dụng của bạn sẽ có quyền truy xuất, chỉnh sửa các tệp tin đặt trong thư mục data của ứng dụng. Tham khảo thêm phần “Lưu trữ dữ liệu” để biết cách đọc ghi tệp tin.

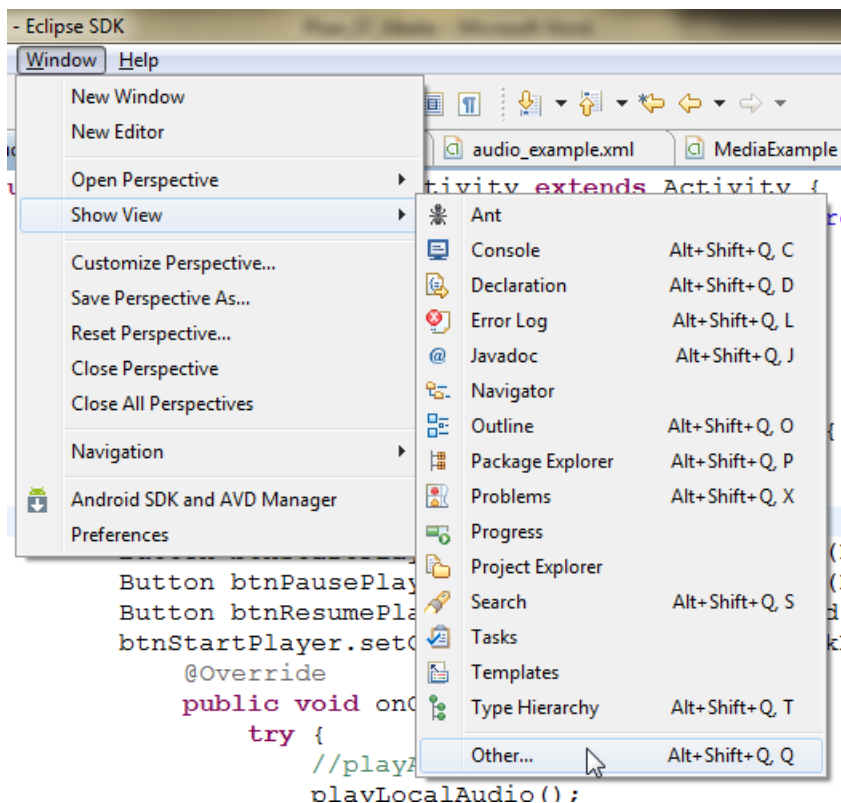
Có sự khác biệt giữa các tệp tin media đặt trong thư mục `res/raw` và thư mục data của ứng dụng. Thư mục `res/raw` là một phần vật lý của tệp tin cài đặt có phần mở rộng `.apk`, các tệp tin đặt trong đây là tệp tin tĩnh, không thể thay đổi sau khi ứng dụng đã được xuất thành tệp tin cài đặt có phần mở rộng `.apk`. Nội dung của thư mục data của ứng dụng thì có thể thay đổi được.

1.2 Chơi tệp tin âm thanh đặt trong thẻ nhớ SD

Để chơi một tệp tin media đặt trong thẻ nhớ SD của việc đầu tiên là chép tệp tin âm thanh lên thẻ nhớ máy ảo:

Bước 01: Mở File Explorer để chép tệp tin âm thanh lên thẻ nhớ của máy ảo:

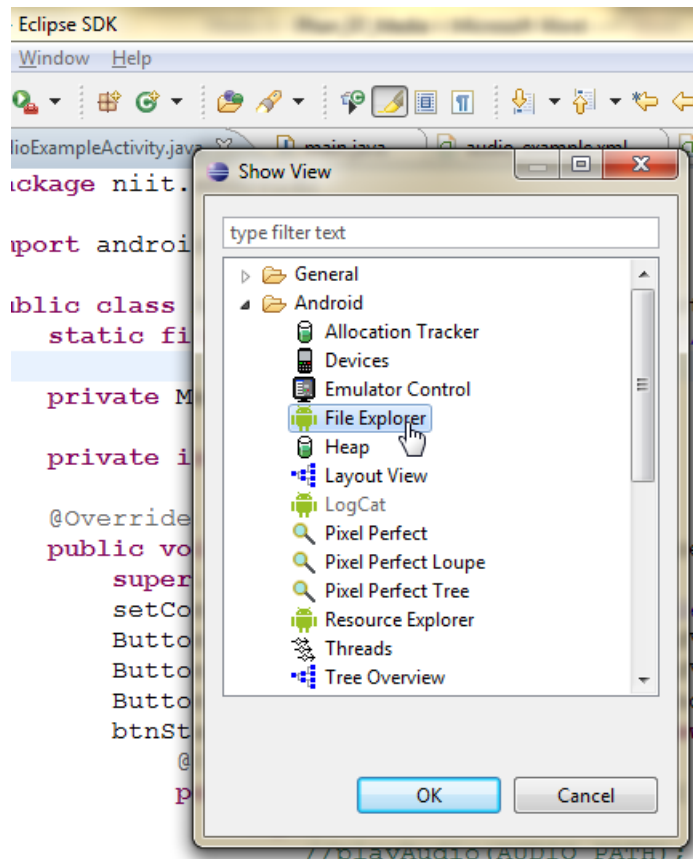
Chọn menu `Windows>Show View>Other`



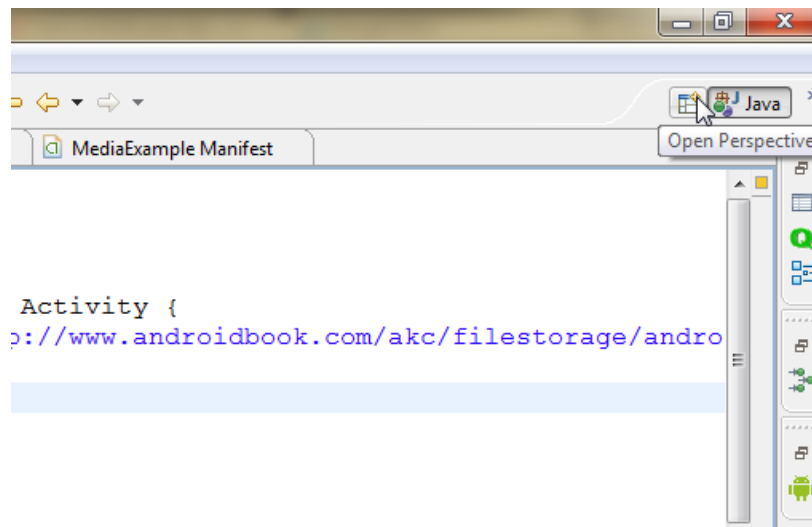
Chọn `Android>File Explorer`:



Phát triển ứng dụng Smartphone – Android



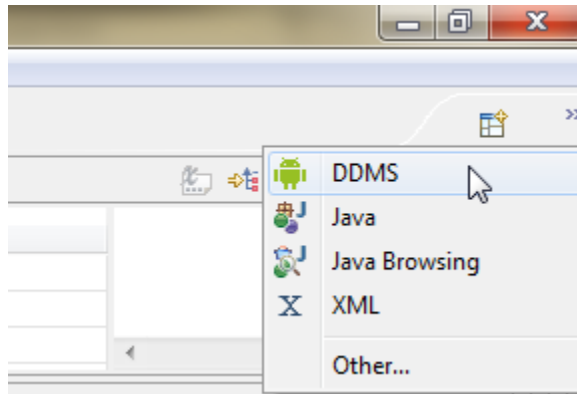
Click chọn “Open Perspective”



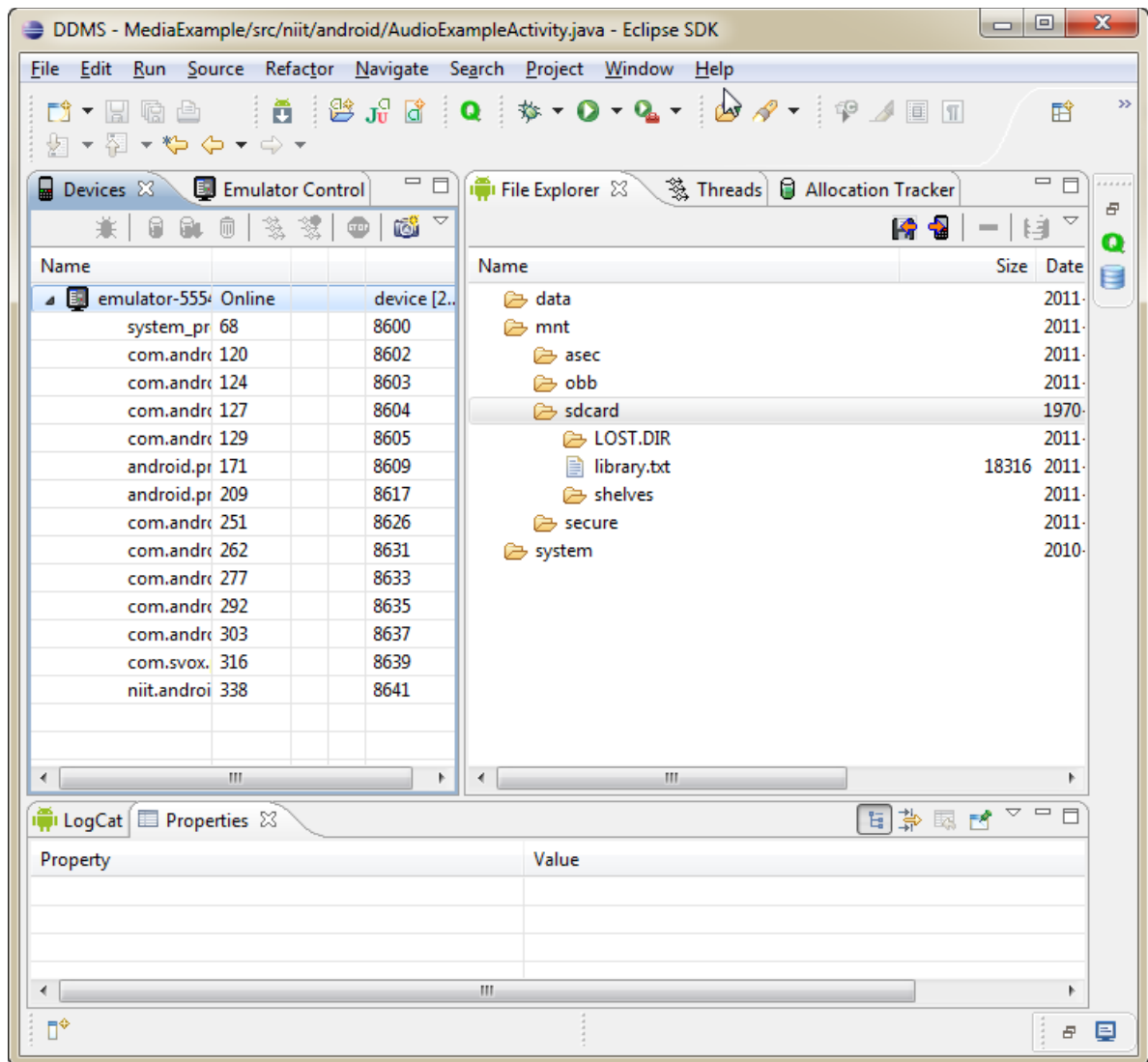
Chọn “DDMS”



Phát triển ứng dụng Smartphone – Android



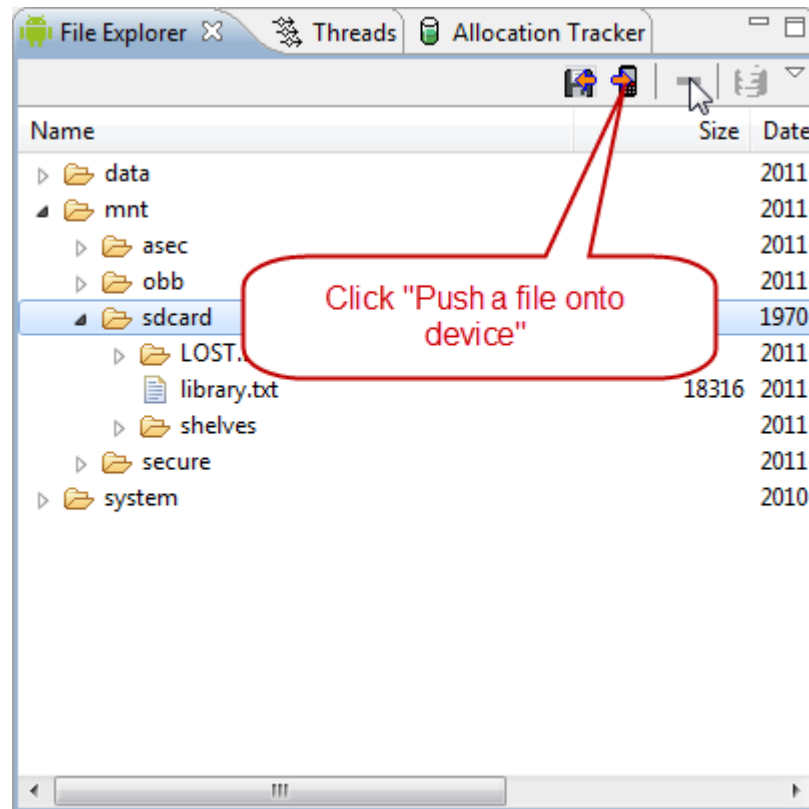
Chọn máy ảo liệt kê ở tab “Devices” và dùng “File Explorer” để mở thư mục sdcard như trong hình:



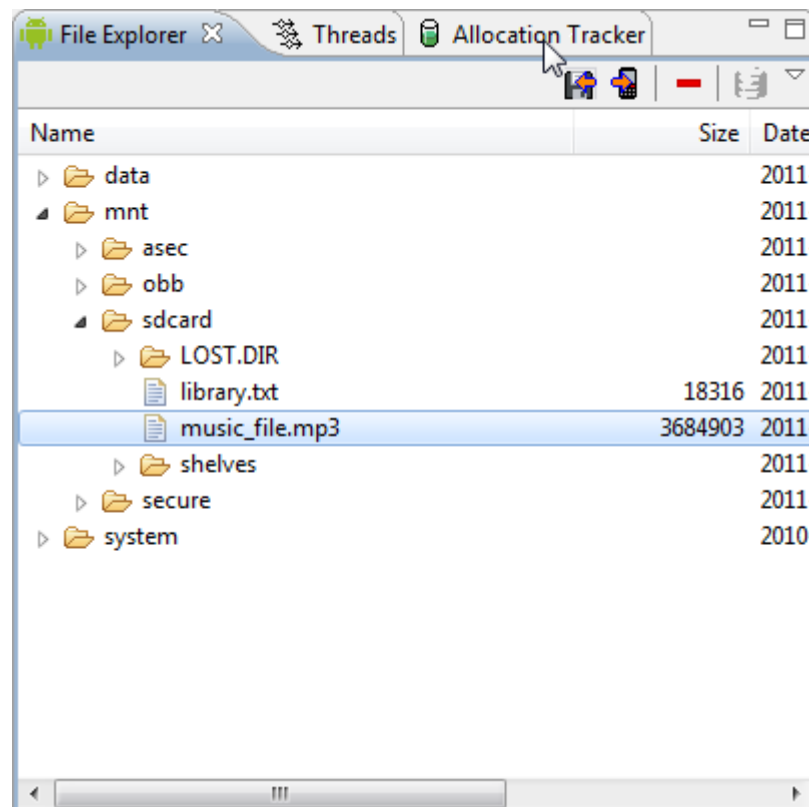
Bước 02: Push một tệp tin media vào thư mục sdcard của máy ảo



Phát triển ứng dụng Smartphone – Android



Chỉ đường dẫn đến tệp tin media để đưa tệp tin vào thẻ sd của máy ảo:





Phát triển ứng dụng Smartphone – Android

Để chơi tệp tin audio kể trên ta chỉ cần thay đổi giá trị của biến AUDIO_PATH của phương thức playAudio() thành:

```
static final String AUDIO_PATH = "/sdcard/music_file.mp3";
```

2 Chơi tệp tin Video

Tạo tệp tin giao diện video_example.xml có nội dung như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center_vertical|center_horizontal">
    <VideoView
        android:id="@+id/videoView"
        android:layout_width="200px"
        android:layout_height="200px" />
</LinearLayout>
```

Tạo mới một activity với nội dung như sau:

```
package niit.android;

import android.app.Activity;
import android.net.Uri;
import android.os.Bundle;
import android.widget.MediaController;
import android.widget.VideoView;

public class VideoExampleActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.setContentView(R.layout.video_example);
        VideoView videoView =
        (VideoView) this.findViewById(R.id.videoView);
        MediaController mc = new MediaController(this);
        videoView.setMediaController(mc);

        videoView.setVideoURI(Uri.parse("http://www.androidbook.com
        /akc/filestorage/android/documentfiles/3389/movie.mp4"));
        videoView.requestFocus();
        videoView.start();
    }
}
```




Phát triển ứng dụng Smartphone – Android

Trong đoạn activity trên, ta sử dụng VideoView để hiển thị video. Nhưng thay vì tự tạo các nút nhấn để điều khiển video đó, ta tạo mediaController để cung cấp các nút nhấn đó cho chúng ta. Trong đoạn code, ta gán MediaController cho videoView bằng cách gọi setMediaController() để cho phép play, pause, and seek-to. Nếu bạn muốn điều khiển bằng các nút nhấn của mình, bạn có thể gọi hàm videoView.start(), videoView.pause(), videoView.stopPlayback(), và videoView.seekTo().

Ngoài ra bạn có thể play video trực tiếp bằng MediaPlayer. Nếu bạn trở lại vd về play audio ở trên và sửa đường dẫn AUDIO_PATH đến 1 video, bạn vẫn sẽ nghe được video này nhưng bạn không thể thấy được hình ảnh.

2.1 Chơi tệp tin video từ thẻ SD

Trong khi MediaPlayer sử dụng hàm setDataSource() thì VideoView sử dụng hàm setVideoPath() hoặc setVideoURI(). Giả sử bạn đặt một tệp tin video vào thẻ nhớ SD như cách đã trình bày ở phía trên và thay đổi code ở trên bằng cách thay hàm setVideoURI() thành setVideoPath(). Và bỏ đường dẫn đến tệp tin media vào hàm setVideoPath() như một tham số. Khi bạn chạy lại ứng dụng, bạn sẽ nghe và nhìn thấy video trên VideoView.

```
videoView.setVideoPath("/sdcard/movie.mp4");
```

Tuy nhiên, bạn cũng có thể sử dụng setVideoURI() để lấy dữ liệu từ SDcard và cho kết quả tương tự như setVideoPath:

```
videoView.setVideoURI(Uri.parse("file:///sdcard/movie.mp4"));
```

3 Một số lưu ý khi làm việc với MediaPlayer và ViewView

Một khi đã gán data source cho một đối tượng MediaPlayer thì ta không dễ dàng thay đổi nó với một data source khác. Thay vào đó tốt nhất bạn nên tạo một MediaPlayer mới hoặc gọi phương thức MediaPlayer.reset() để khởi tạo lại đối tượng MediaPlayer.

Sau khi gọi phương thức MediaPlayer.prepare() bạn có thể gọi phương thức getCurrentPosition(), getDuration() và isPlaying() để kiểm tra trạng thái của MediaPlayer. Ngoài ra bạn còn có thể gọi phương thức setLooping() và setVolume() để yêu cầu MediaPlayer chơi lặp đi lặp lại tệp tin media của bạn cũng như thay đổi âm lượng.

Sau khi gọi phương thức Media.start() bạn có thể gọi các phương thức MediaPlayer.pause(), MediaPlayer.stop() và MediaPlayer.seekTo().

Mỗi MediaPlayer sẽ tự tạo một luồng mới để chơi tệp tin media bạn chỉ định, chính vì vậy cần phải gọi phương thức MediaPlayer.release() khi không còn sử dụng MediaPlayer. Nếu bạn sử dụng VideoView để trình diễn một tệp tin video thì VideoView sẽ làm việc này cho bạn.

Có thể sử dụng MediaPlayer để chơi tệp tin video, tất nhiên là ta chỉ có thể nghe được âm thanh của tệp tin video này mà không xem được hình ảnh.



4 Ghi âm thanh

Tạo mới một tệp tin audio_record.xml với nội dung như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- This file is /res/layout/record.xml -->
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:id="@+id/bgnBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Begin Recording" />
    <Button
        android:id="@+id/stpBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Stop Recording" />

    <Button
        android:id="@+id/playRecordingBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Play Recording" />

    <Button
        android:id="@+id/stpPlayingRecordingBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Stop Playing Recording" />

</LinearLayout>
```

Tạo mới một Activity với nội dung:

```
package niit.android;

import java.io.File;
import android.app.Activity;
import android.media.MediaPlayer;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
```



Phát triển ứng dụng Smartphone – Android

```
public class AudioRecorderActivity extends Activity {
    private MediaPlayer mediaPlayer;

    private MediaRecorder recorder;

    private static final String OUTPUT_FILE =
"/sdcard/recordaudio.3gpp";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.audio_record);
        Button startBtn = (Button)findViewById(R.id.bgnBtn);
        Button endBtn = (Button)findViewById(R.id.stpBtn);
        Button playRecordingBtn =
(Button)findViewById(R.id.playRecordingBtn);
        Button stopPlayingRecordingBtn =
(Button)findViewById(R.id.stpPlayingRecordingBtn);
        startBtn.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    beginRecording();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
        endBtn.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    stopRecording();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
        playRecordingBtn.setOnClickListener(new
OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    playRecording();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```



Phát triển ứng dụng Smartphone – Android

```
    }
    }
    });
    stopPlayingRecordingBtn.setOnClickListener(new
OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            stopPlayingRecording();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    });
}

private void beginRecording() throws Exception {
    killMediaRecorder();
    File outFile = new File(OUTPUT_FILE);
    if (outFile.exists()) {
        outFile.delete();
    }
    recorder = new MediaRecorder();
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_G
PP);
    recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB)
;

    recorder.setOutputFile(OUTPUT_FILE);
    recorder.prepare();
    recorder.start();
}

private void stopRecording() throws Exception {
    if (recorder != null) {
        recorder.stop();
    }
}

private void killMediaRecorder() {
    if (recorder != null) {
        recorder.release();
    }
}

private void killMediaPlayer() {
    if (mediaPlayer != null) {
```



Phát triển ứng dụng Smartphone – Android

```
        try {
            mediaPlayer.release();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void playRecording() throws Exception {
        killMediaPlayer();
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setDataSource(OUTPUT_FILE);
        mediaPlayer.prepare();
        mediaPlayer.start();
    }

    private void stopPlayingRecording() throws Exception {
        if (mediaPlayer != null) {
            mediaPlayer.stop();
        }
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        killMediaRecorder();
        killMediaPlayer();
    }
}
```

Việc ghi âm thanh trong Android sẽ thông qua một đối tượng MediaRecorder. Tuy nhiên trước tiên ta cần bổ xung quyền ghi âm thanh, và quyền ghi tệp tin xuống thẻ nhớ cho ứng dụng bằng cách bổ xung tệp AndroidManifest.xml:

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Phương thức beginRecording() có các lệnh sau:

```
recorder = new MediaRecorder();
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
recorder.setOutputFile(OUTPUT_FILE);
recorder.prepare();
recorder.start();
```




Trước tiên là tạo mới một đối tượng `MediaRecorder`. Sau đó gọi phương thức `setAudioSource()` để gán nguồn âm cần thu. Nguồn âm này có thể là từ mic, từ cuộc gọi đến, từ tệp tin nhạc đang phát... Phương thức `setOutputFormat()` để thiết lập định dạng âm thanh và phương thức `setAudioEncoder()` để thiết lập cách mã hóa âm thanh và `setOutPutFile()` để gán đường dẫn để lưu tệp tin âm thanh xuống.

Ngoài ra còn các phương thức hữu dụng là:

- `setMaxDuration()`: Dùng để gán thời gian tối đa của tệp tin âm thanh ghi thành công.
- `setMaxFileSize()`: Gán kích thước tối đa của tệp tin âm thanh ghi thành công.

5 Ghi video

Tạo mới một tệp tin `video_record.xml` với nội dung như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <Button android:id="@+id/bgnBtn"
    android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="Begin
Recording"
        android:enabled="false" />

    <Button android:id="@+id/stpBtn"
    android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="Stop
Recording" />

    <Button android:id="@+id/playRecordingBtn"
    android:layout_width="fill_parent"
        android:layout_height="wrap_content" android:text="Play
Recording" />

    <Button android:id="@+id/stpPlayingRecordingBtn"
        android:layout_width="fill_parent"
    android:layout_height="wrap_content"
        android:text="Stop Playing Recording" />

    <RelativeLayout android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center">
```



Phát triển ứng dụng Smartphone – Android

```
<VideoView android:id="@+id/videoView"
android:layout_width="176px"
    android:layout_height="144px" />

</RelativeLayout>
</LinearLayout>
```

Tạo mới một Activity với nội dung như sau:

```
package niit.android;

import java.io.File;
import android.app.Activity;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.util.Log;
import android.view.SurfaceHolder;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.VideoView;

public class VideoRecorderActivity extends Activity implements
SurfaceHolder.Callback {
    private MediaRecorder recorder = null;

    private static final String OUTPUT_FILE =
"/sdcard/videooutput.mp4";

    private static final String TAG = "RecordVideo";

    private VideoView videoView = null;

    private Button startBtn = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        startBtn = (Button) findViewById(R.id.bgnBtn);
        Button endBtn = (Button) findViewById(R.id.stpBtn);
        Button playRecordingBtn =
(Button) findViewById(R.id.playRecordingBtn);
        Button stpPlayingRecordingBtn =
(Button) findViewById(R.id.stpPlayingRecordingBtn);
        videoView =
(VideoView) this.findViewById(R.id.videoView);
```



Phát triển ứng dụng Smartphone – Android

```
final SurfaceHolder holder = videoView.getHolder();
holder.addCallback(this);

holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
startBtn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            beginRecording(holder);
        } catch (Exception e) {
            Log.e(TAG, e.toString());
            e.printStackTrace();
        }
    }
});
endBtn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            stopRecording();
        } catch (Exception e) {
            Log.e(TAG, e.toString());
            e.printStackTrace();
        }
    }
});
playRecordingBtn.setOnClickListener(new
OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            playRecording();
        } catch (Exception e) {
            Log.e(TAG, e.toString());
            e.printStackTrace();
        }
    }
});
stpPlayingRecordingBtn.setOnClickListener(new
OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            stopPlayingRecording();
        } catch (Exception e) {
            Log.e(TAG, e.toString());
            e.printStackTrace();
        }
    }
});
```



Phát triển ứng dụng Smartphone – Android

```
        }
    }
    });
}

@Override
public void surfaceCreated(SurfaceHolder holder) {
    startBtn.setEnabled(true);
}

@Override
public void surfaceDestroyed(SurfaceHolder holder) {
}

@Override
public void surfaceChanged(SurfaceHolder holder, int
format, int width, int height) {
    Log.v(TAG, "Width x Height = " + width + "x" +
height);
}

private void playRecording() {
    MediaController mc = new MediaController(this);
    videoView.setMediaController(mc);
    videoView.setVideoPath(OUTPUT_FILE);
    videoView.start();
}

private void stopPlayingRecording() {
    videoView.stopPlayback();
}

private void stopRecording() throws Exception {
    if (recorder != null) {
        recorder.stop();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (recorder != null) {
        recorder.release();
    }
}
```



Phát triển ứng dụng Smartphone – Android

```
private void beginRecording(SurfaceHolder holder) throws
Exception {
    if (recorder != null) {
        recorder.stop();
        recorder.release();
    }
    File outFile = new File(OUTPUT_FILE);
    if (outFile.exists()) {
        outFile.delete();
    }
    try {
        recorder = new MediaRecorder();
        recorder.setVideoSource(MediaRecorder.VideoSource
            .CAMERA);
        recorder.setAudioSource(MediaRecorder.AudioSource
            .MIC);
        recorder.setOutputFormat(MediaRecorder.OutputForm
            at.MPEG_4);
        recorder.setVideoSize(176, 144);
        recorder.setVideoFrameRate(15);
        recorder.setVideoEncoder(MediaRecorder.VideoEncod
            er.MPEG_4_SP);
        recorder.setAudioEncoder(MediaRecorder.AudioEncod
            er.AMR_NB);
        recorder.setMaxDuration(5000); // limit to 5
seconds

        recorder.setPreviewDisplay(holder.getSurface());
        recorder.setOutputFile(OUTPUT_FILE);
        recorder.prepare();
        recorder.start();
    } catch (Exception e) {
        Log.e(TAG, e.toString());
        e.printStackTrace();
    }
}
```

Ta cũng lần lượt gọi phương thức `setVideoSource()` để thiết lập nguồn video cần thu (Camera...), `setAudioSource()` để thiết lập nguồn âm thanh cần thu, `setOutputFormat()` để thiết lập định dạng xuất ra (MP4), `setVideoSize()` để gán kích thước tệp video cần ghi, `setVideoFrameRate()` để thiết lập số khung hình trên giây, `setAudioEncoder()` để thiết lập cách mã hóa âm thanh, `setMaxDuration()` để thiết lập số giây tối đa ghi lại, phương thức `setPreviewDisplay()` để xem trước phần video đang quay.