Fpt University

FPT POLYTECHNIC

ANDROID NÂNG CAO

BÀI 8: CAMERA - MEDIA

www.poly.edu.vn

Mục tiêu

- Kết thúc bài học này bạn có khả năng
  - Sử dụng Camera
  - Sử dụng Media

NỘI DUNG

- Phần I: Camera
  - Picture
  - Video
- Phần II: Media
  - Playing Audio
  - MediaPlayer

Fpt University

FPT POLYTECHNIC

BÀI 8: CAMERA - MEDIA

PHẦN I: CAMERA

www.poly.edu.vn

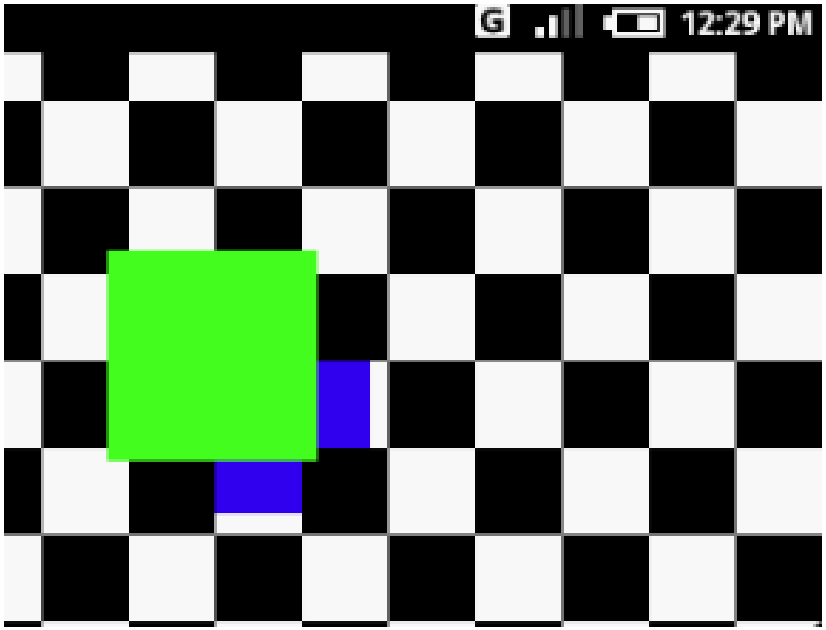
CAMERA

- Lớp **Camera** được sử dụng để thiết lập camera, bắt đầu hoặc dừng sử dụng camera và preview ảnh
- Để truy cập camera, bạn phải khai báo quyền CAMERA trong Android Manifest
- Thêm phần tử <uses-feature> để khai báo các tính năng của camera mà ứng dụng hỗ trợ

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
```

CAMERA

- Camera trên Emulator không hoạt động
  - Preview: di chuyển khối hình
  - Chụp ảnh luôn luôn hiển thị cùng một bức ảnh trắng
- Để test ứng dụng Camera trên Emulator phải kích hoạt camera trên Emulator



Các bước sử dụng Camera

- Sử dụng **open(int)** để nhận instance của Camera
- Lấy thông tin các thiết lập mặc định sử dụng **getParameters()**
- Nếu cần thiết, thay đổi đối tượng trả về Camera.Parameters và gọi **setParameters(Camera.Parameters)**
- Nếu cần thiết, gọi **setDisplayOrientation(int)**
- Quan trọng: truyền SurfaceHolder (đã được khởi tạo đầy đủ) tới **setPreviewDisplay(SurfaceHolder)**. Không có Surface, camera không thể bắt đầu preview

CAMERA

- Gọi **startPreview()** để bắt đầu cập nhật preview serface. Preview phải bắt đầu trước khi bạn chụp ảnh
- Khi bạn muốn, gọi **takePicture(Camera.ShutterCallback, Camera.PictureCallback, Camera.PictureCallback,Camera.PictureCallback)** để capture ảnh. Đợi callback cung cấp ảnh chụp thật
- Sau khi chụp ảnh, preview display sẽ dừng. Để chụp thêm ảnh gọi **startPreview()** lần nữa
- Gọi **stopPreview()** để dừng cập nhật preview surface
- Quan trọng: gọi **release()** để giải phóng camera được sử dụng bởi ứng dụng khác. Ứng dụng nên giải phóng Camera ngay lập tức trong **onPause()** (và mở lại trong **onResume()**)

Video recording

- Lấy thông tin và khởi tạo Camera và bắt đầu preview như miêu tả ở trên
- Gọi **unlock()** để cho phép media process thao tác với camera
- Truyền camera tới **setCamera(camera)**. Sử dụng **MediaRecorder** để quay phim
- Khi kết thúc quay, gọi **reConnect()** để khóa camera
- Nếu muốn, khởi tạo lại preview và tiếp tục chụp ảnh hoặc quay video
- Gọi stopPreview() và release() giống như trên

DEMO

Sử dụng Camera

Fpt University

FPT POLYTECHNIC

BÀI 8: CAMERA - MEDIA

PHẦN II: MEDIA

www.poly.edu.vn

Playing Audio

- Android cung cấp 2 API để play audio
  - SoundPool
  - MediaPlayer
- SoundPool có thể được sử dụng để play các clip có kích thước nhỏ. Có thể lặp âm thanh và play một số file đồng thời
- SoundPool chỉ nên bật các file âm thanh có kích thước không quá 1MB
- SoundPool tải các file không đồng bộ
- Từ Android API8, có thể kiểm tra việc tải file đã hoàn thành chưa sử dụng OnLoadCompleteListener

Playing Audio

- Phone volume có thể được cấu hình để điều khiển audio stream, ví dụ trong khi gọi volume button cho phép tăng/giảm dung lượng
- MediaPlayer phù hợp hơn cho các movie và music có dung lượng lớn

Playing Audio

```
MediaPlayer mp = MediaPlayer.create(context, R.raw.sound);
mp.start();
... mp.stop();
```

- Nếu stop, phải gọi mp.reset() và mp.prepare() trước khi gọi mp.start() lần nữa
- Có thể tạm dừng mp.pause() rồi quay trở lại sử dụng mp.start()
- Đảm bảo gọi mp.release() khi kết thúc

Playing File hoặc Stream

- Có thể truyền đối tượng Uri trỏ tới local file hoặc internet stream
- Làm việc với HTTP/RTSP stream
- Làm việc với định dạng file nhất định

Playing Video

- Tương tự như audio, ngoại trừ phải copy resource trước
- Bổ sung: truyền **SurfaceHolder** tới player có thể sinh video
- Giống như cửa sổ preview của Camera
- **getHolder()** của **SurfaceView**
- Copy file tới emulator card để test

Media Event

- Media Player có thể ném ra một ngoại lệ: ví dụ format không được chấp nhận
- Media Player hỗ trợ một số listerer để tương tác với player
  - OnErrorListener
  - OnBufferingUpdateListener

```
public class MyService extends Service implements MediaPlayer.OnErrorListener {
    MediaPlayer mMediaPlayer;

    public void initMediaPlayer() {
        // ...initialize the MediaPlayer here...

        mMediaPlayer.setOnErrorListener(this);
    }

    @Override
    public boolean onError(MediaPlayer mp, int what, int extra) {
        // ... react appropriately ...
        // The MediaPlayer has moved to the Error state, must be reset!
    }
}
```

Sử dụng wake lock

- Khi thiết kế ứng dụng dùng để chơi nhạc dưới dạng background, thiết bị có thể sleep khi service đang chạy bởi vì hệ thống Android cố gắng tiết kiệm pin khi thiết bị đang sleep
- Nếu service đang chơi một bản nhạc, bạn có thể muốn ngăn cản hệ thống can thiệp vào service nghe nhạc
- Để đảm bảo service vẫn tiếp tục chạy, bạn phải sử dụng wake lock. Wake lock là một cách để báo cho hệ thống biết là một số chức năng vẫn chạy kể cả khi phone ở trạng thái idle

Sử dụng wake lock

- Để đảm bảo rằng CPU tiếp tục chạy khi MediaPlayer đang chơi nhạc, gọi phương thức **setWakeMode()** khi khởi tạo MediaPlayer
- Khi đó MediaPlayer sẽ bị khóa khi chơi nhạc và giải phóng khóa khi tạm dừng hoặc dừng chơi nhạc

```
mMediaPlayer = new MediaPlayer();
// ... other initialization here ...
mMediaPlayer.setWakeMode(getApplicationContext(), PowerManager.PARTIAL_WAKE_LOCK);
```

Sử dụng wake lock

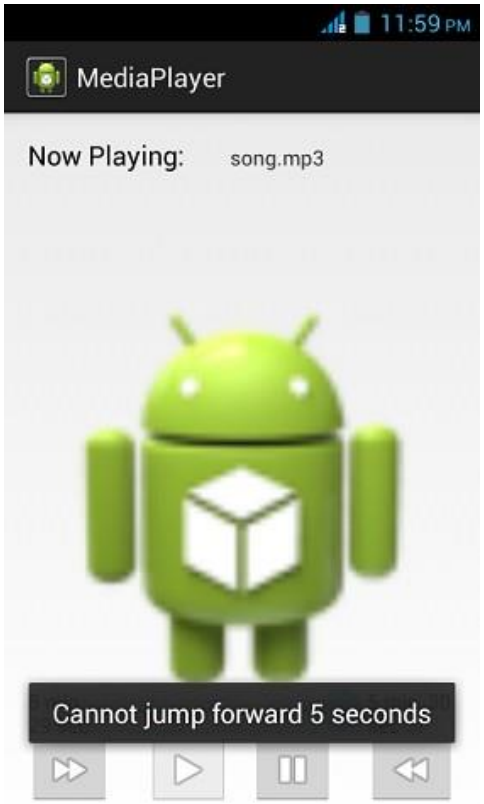
- Tuy nhiên, wake look chỉ đảm bảo là CPU vẫn không sleep
- Nếu bạn muốn stream một bản nhạc qua Wifi, bạn có thể cần giữ cho Wifi không bị tắt. Do đó, bạn cần giữ WifiLock và bạn phải lấy thông tin WifiLock và giải phóng một cách thủ công
- Do đó, khi bạn bắt đầu chuẩn bị MediaPlayer với một URL, bạn nên tạo và lấy thông tin Wifi lock

```
WifiLock wifiLock = ((WifiManager) getSystemService(Context.WIFI_SERVICE))
        .createWifiLock(WifiManager.WIFI_MODE_FULL, "mylock");

wifiLock.acquire();
```

Định dạng được hỗ trợ

- Core Format hỗ trợ trên mọi thiết bị
  - Thiết bị có thể chọn hỗ trợ thêm các định dạng khác
  - Một số không được bổ sung vào SDK < 2.2, do đó cần test trên các thiết bị và emulator cũ
- Định dạng chung:
  - .3gp , .mp4.m4a
  - .mp3, .ogg, .wav audio
  - jpg, gif, png, bmp images



DEMO

Thao tác với sound trong ứng dụng Android

TỔNG KẾT NỘI DUNG BÀI HỌC

- Phần I: Camera
  - Picture
  - Video
- Phần II: Media
  - Playing Audio
  - MediaPlayer


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
Cảm ơn