

# Android alkalmazásfejlesztés

Kamera használata

Kép rögzítése

YUV színrendszer megismerése

Videó rögzítése

Arcdetektálás

OE-NIK

2011. december 4.

**Sicz-Mesziár János**

sicz-mesziar.janos@  
nik.uni-obuda.hu



# Kamera jellemzők

## Kamera kezelésének nehézségei:

- Nagy mennyiségű adat :

- 3 MP ~ 4 MB adat



- 5 MP ~ 😊

- 8 MP ~ 😊😊

## ◎ Sok jellemző és beállítási lehetőség

- Felbontás, képarány
- FPS (frame per sec)
- Képminőség, ISO
- Kép tulajdonságai: kontraszt, saturáció, fényerő, ...
- Effektek: szépia, szürkeárnyalat, negatív, ...

## ◎ És még új/más dimenziók:

- Vaku, Zoom, 2 kamera, Autófókusz, ...

TYPE	FILESIZE	DESCRIPTION
RAW12	4.7 MB	(uncompressed 12 bits/pixel - Bayer mask)
RAW10	3.9 MB	(uncompressed 10 bits/pixel - Bayer mask)
RAW8	3.1 MB	(uncompressed 8 bits/pixel - Bayer mask)
BMP	9.4 MB	(uncompressed RGB 24bit/pixel)
TIFF	18.9 MB	(uncompressed CMYK 48bit/pixel)
OPENEXR	25.2 MB	(uncompressed RGBA 4 * 16bit float/pixel)
FILE COMPRESSION ESTIMATIONS		
JPG100	1.0 MB	(100% quality - 24bit/pixel)
JPG90	504.4 KB	(90% quality - 24bit/pixel)
GIF	1.1 MB	(compressed - 8bit/pixel)
PNG	2.5 MB	(lossless compressed - 24bit/pixel)

**Kérdés:**  
**mit támogat a kamera?**

**Kérdés:**  
**mit támogat az adott  
Android verzió?**

# Kamera előnézeti kép

## 1. AndroidManifest.xml-ben hozzáférés kérése:

Permissions > Uses permission : `android.permission.CAMERA`

Manifest Extras > Uses feature :

`android.hardware.camera`

`android.hardware.camera.autofocus`

⊙ Kell egy `SurfaceView` leszármazott osztály.

⊙ `SurfaceHolder` biztosítja a felületet a kamerának.

■ `surfaceCreated()` - felület kiépítése → kamera ON

■ `surfaceChanged()` - vmi. változik → paraméter SET

■ `surfaceDestroyed()` - felület bezár → kamera OFF

⊙ A paraméterek beállítása, kicsit macerás:

- Más a hardveres támogatottság



`cam.getParameters().getSupported...()`

- **Mindig le kell kérni a támogatott tulajdonságokat!**

# Autófókusz

- ⊙ Az automatikus fókusz beállításához implementálni kell egy `AutoFocusCallback`-et:

```
AutoFocusCallback focus = new AutoFocusCallback() {  
    public void onAutoFocus(boolean success, Camera cam) {  
        // Pl.: egy fotó készítése  
    }  
};
```

Akkor hívódik meg, ha az autófókusz befejeződött.

- ⊙ Implementált `AutoFocusCallback` megadása:

```
...  
cam.setParameters(params);  
cam.startPreview();  
cam.autoFocus(focus);
```

**Fontos:** az autófókusz indítása csakis **`startPreview()`** után lehet!

# Előnézeti kép elő-feldolgozása

⊙ Az előnézeti képek feldolgozhatóak, képkockánként.

- Pl.: valós idejű detektálások – arcdetektálás

⊙ Megvalósítása:

- `surfaceCreated()` -ben beállítjuk a `PreviewCallback`-et

```
cam.setPreviewCallback(this);  
cam.setPreviewDisplay(holder);
```

- Osztályunk implementálja ezt a `PreviewCallback`-et:

```
public void onPreviewFrame(byte[] data, Camera camera){  
    /* adatok data[] tömbben, params.setPreviewFormat() -nak megfelelően */  
}
```

⊙ **Probléma:** Nagy adatmennyiség → GC takarít a memóriában, **sokáig**. *(Több pixel → gyakoribb GC művelet)*

Time	pid	tag	Message
04-06 14:59:25.581	D 1...	dalvikvm	GC freed 5 objects / 345720 bytes in 81ms
04-06 14:59:26.061	D 1...	dalvikvm	GC freed 5 objects / 345720 bytes in 80ms
04-06 14:59:26.361	D 1...	dalvikvm	GC freed 5 objects / 345720 bytes in 88ms
04-06 14:59:26.631	D 1...	dalvikvm	GC freed 5 objects / 345720 bytes in 98ms
04-06 14:59:26.881	D 1...	dalvikvm	GC freed 5 objects / 345720 bytes in 102ms



25fps esetén:  
 $1000\text{ms}/25 =$   
**40ms**

**Megoldás:** `addCallbackBuffer()` használata, de csak 2.2 óta.  
Egyébként hack, de csak 2.1-re.

# YUV színrendszer

- Előnézeti kép „byte stream” formátuma beállítható, de csak a szoftver által támogatott formátumokra.
- Alapértelmezett: NV21. Ez egy YUV420-as formátum:

Single Frame YUV420:



Position in byte stream:



- Konverzió hogyan?

Vannak egyenletek. Van megírt eljárás. YuvImage class 2.2 óta.

YCbCr néven is ismert:

Y = luminancia (világosság)

U = Cb = (Y-B), krominancia (szín)

V = Cr = (Y-R), krominancia (szín)

## Miért ez?

- Fekete-fehér analóg adások miatt hozták létre
- Ma már a digitális kódolásban, inkább tömörítési eljárásokban használják: JPEG, MPEG

# Fotó készítése

## ◎ Kép készítéséhez 3 Callback implementálása kell:

- `ShutterCallback shutter = new ShutterCallback() {...}`
- `PictureCallback raw = new PictureCallback() {...}`
- `PictureCallback jpeg = new PictureCallback() {...}`

## ◎ Képrögzés az összes Callback megadásával:

```
csiizBtn.setOnClickListener(new OnClickListener() {  
    public void onClick(View v) {  
        if(cam != null)  
            cam.takePicture(shutter, raw, jpeg);  
    }  
});
```

## ◎ Példa JPEG mentésre:

```
private PictureCallback jpeg = new PictureCallback() {  
    public void onPictureTaken(byte[] data, Camera camera) {  
        FileOutputStream fos = new  
            FileOutputStream("/sdcard/cam.jpg");  
        fos.write(data);  
        fos.close();  
    }  
};
```

# Videó rögzítés

## ◎ MediaRecorder használatával:

```
MediaRecorder rec = new MediaRecorder();  
rec.setAudioSource(MediaRecorder.AudioSource.DEFAULT);  
rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);
```

## ◎ További adatok:

```
CamcorderProfile cpHigh = CamcorderProfile  
    .get(CamcorderProfile.QUALITY_HIGH);  
rec.setProfile(cpHigh);  
rec.setOutputFile("/sdcard/videocapture_example.mp4");  
rec.setMaxDuration(50000); // 50 mp  
rec.setMaxFileSize(5000000); // Kb. 5 MB
```

## ◎ Rögzítés indítása/megállítása

```
rec.setPreviewDisplay(holder.getSurface());  
try{  
    rec.prepare();  
}catch(...) {...} surfaceCreated()  
rec.start(); Button onClick()  
rec.stop();  
rec.release(); surfaceDestroyed()
```



# Arc detektálás

## ◎ Mit lehet detektálni?

- Arc detektálása fotón, kamera előnézeti képen
- Szemek elhelyezkedése (MidPoint)
- Szemek közti távolság (Distance)



## ◎ Gyakorlatban:

```
byte NUM_FACE = 10;
FaceDetector fd = new FaceDetector(picW, picH, NUM_FACE);
FaceDetector.Face allFaces[] = new
    FaceDetector.Face[NUM_FACE];
fd.findFaces(source, allFaces);    // source <- Bitmap
for(FaceDetector.Face f : allFaces){
    PointF eyesMP = new PointF();
    f.getMidPoint(eyesMP);         // Szem-középpont
    f.eyesDistance();             // Szemek közti táv
    f.confidence();               // Megbízhatóság
    // f.pose
}
```