

## ■ 圖形處理

### A. 繪製圖片/圖形 (遊戲、動畫)

(方法一)

```
public class GameView extends View {
    public void onDraw(Canvas canvas) {
        // 繪圖指令 canvas.drawXXX(...);
    }
    public void onMeasure(...) {
    }
}
```

註：若是遊戲、動畫，需要搭配客制化 Thread + **xx.postInvalidate()** 以定期執行 **xx.onDraw()**

### (方法二) **Customized SurfaceView** (高效繪圖處理)

- 取得 **SurfaceView** 物件的 **SurfaceHolder**
  - **sfHolder = SurfaceView.getHolder();**
  - **sfHolder.addCallback(...);**
- 覆寫必要的 **Callback** 方法 ( **SurfaceHolder.Callback** )
  - **surfaceCreated(\*)**
  - **surfaceChanged(...)**
  - **surfaceDestroyed()**
- 自訂 **Drawing Thread (Stand-alone)** - 定期執行 (亦可用 **Timer+TimerTask**)
  - **Canvas canvas = sfHolder.lockCanvas( );**
  - **// 由 canvas 進行 drawing ! 畫完後即可生效(看到輸出成果) !!**
  - **sfHolder.unlockCanvasAndPost(canvas);**
- 相關類別:
  - **Canvas:**
  - **Paint:**
  - **Color:**
  - ...

### B. 貼圖相關

- **Bitmap** **pic = BitmapFactory.decodeResource( getResources(), R.drawable.xxx );**
- **Canvas.drawBitmap( pic, x, y, pen);**

### C. 自訂動畫相關

- **Tweened (漸變) animation** 、 **frame by frame(視框) animation**
  - **alpha-透明度、scale-縮放、translate-轉換位置、rotate-旋轉**
- 定義動畫集 (set): **/res/anim/xxx.xml**
- 載入動畫定義:
  - anim = AnimationUtils.loadAnimation( context, R.anim. xxx);**
- 啟動動畫: **drawableObj.startAnimation( anim );**

## ■ 多媒體處理

- 音訊的播放
  - MediaPlayer**: 單一檔案
  - SoundPool**: 多個檔案 (音效檔案)
- 視訊的播放
  - MediaPlayer** (略)
  - VideoView** (= MediaPlayer + SurfaceView + Controller)

```
String vfile = "android.resource://" + this.getPackageName() + "/" + R.raw.mp4;  
// String vfile = "/sdcard/love_480320.mp4";  
Uri uri = Uri.parse(vfile);  
videoView.setVideoURI(uri);  
// or videoView.setVideoPath("file:///sdcard/love_480320.mp4");  
videoView.setMediaController(new MediaController(this));  
videoView.requestFocus();  
videoView.start();
```

## ■ 利用 MediaPlayer 控制影音的主要步驟：

- 生成 **MediaPlayer** 物件

```
MediaPlayer mp = MediaPlayer.create(context, R.raw.xxx); // 對應內部資源
```

// 或者

```
MediaPlayer mp = new MediaPlayer();
```

```
mp.setDataSource(...);
```

// 設定媒體來源 (mp:idle)

```
mp.prepare();
```

// 載入媒體資料

```
//檔案較大時建議採用: mp.prepareAsync();
```

```
// 採用非同步時, 須加 Listener: mp.setOnPreparedListener(...);
```

- 啟動/開始播放 (必須是 **prepared** 的狀態)

```
mp.start();
```

- 暫停播放

```
mp.pause();
```

- 停止播放

```
mp.stop();
```

//停止後要再重新播放, 必須先呼叫 prepare()

- 釋放資源

```
mp.release();
```

- 利用 **MediaPlayer** 播放遊戲背景音樂時, 必需要搭配 **Activity** 的生命週期的改變, 在不同時間點, Activity.onXXX(...), 進行音效控制(暫停、繼續、停止, ...)。

