

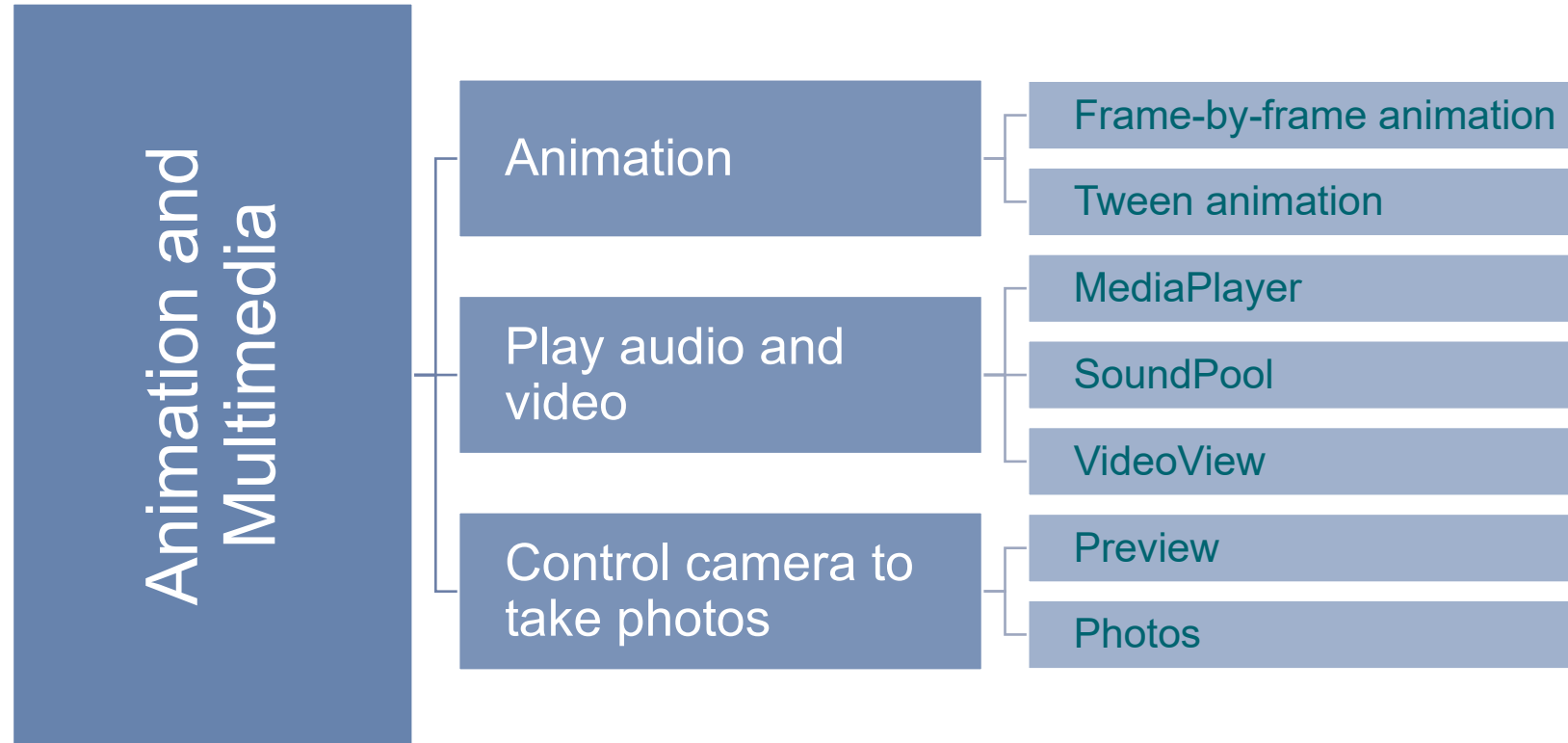
移动应用开发

张昕

zhangxin@cust.edu.cn

2022

Animation and Multimedia



Frame-by-frame animation

- **Sequentially playing images to display animation**

- **Steps**

- **Define a set of image resource in XML resource file**

android:oneshot attribute specifies
whether to repetitively play

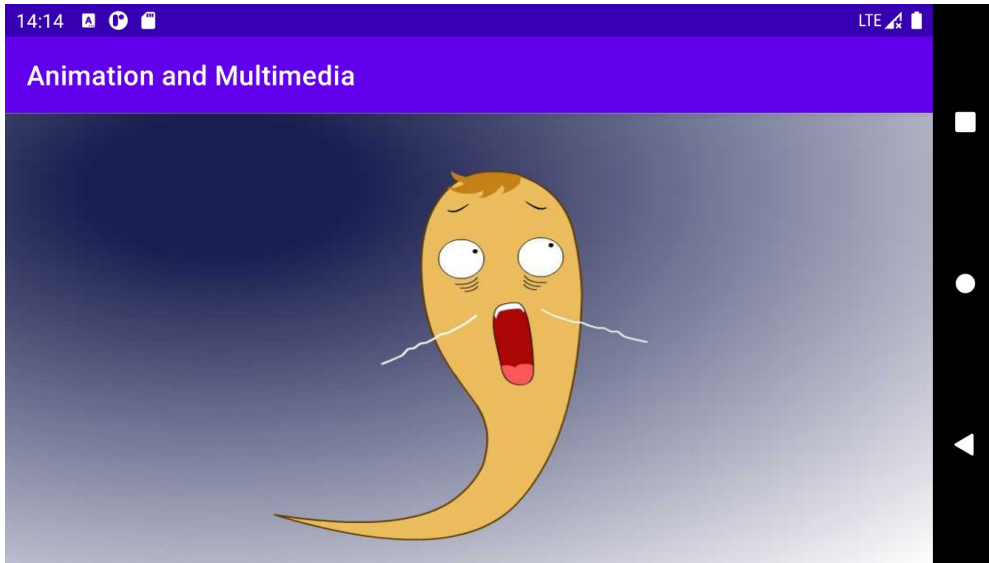
```
<animation-list xmlns:android=http://schemas.android.com/apk/res/android android:oneshot="true|false">  
    <item android:drawable="@drawable/imageResource1" android:duration="integer"/>  
    ...  
    <item android:drawable="@drawable/imageResource1" android:duration="integer"/>  
</animation-list>
```

- **Use the defined image resource as the background of components**

Such as,

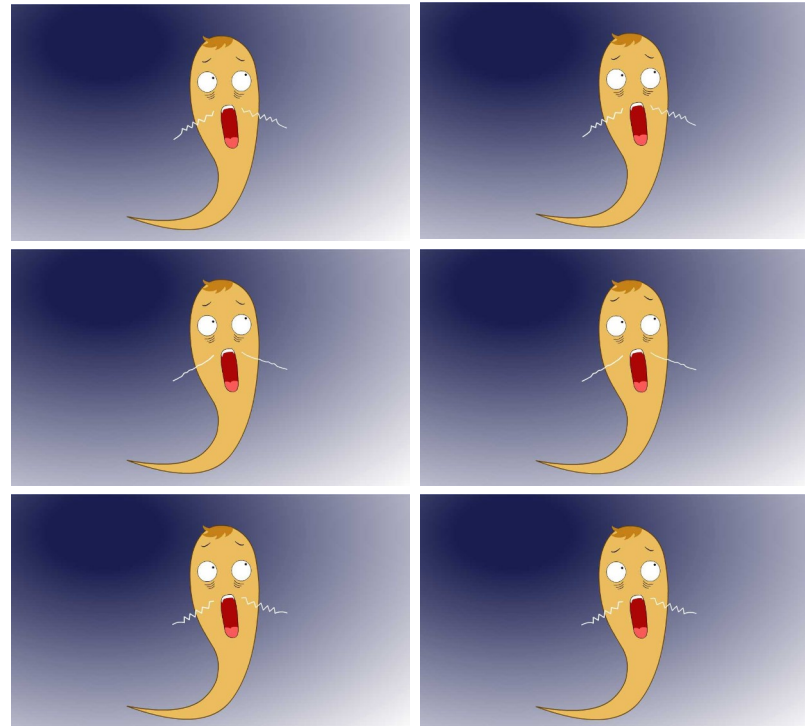
Layout/ImageView → android:background attribute

Small Demo

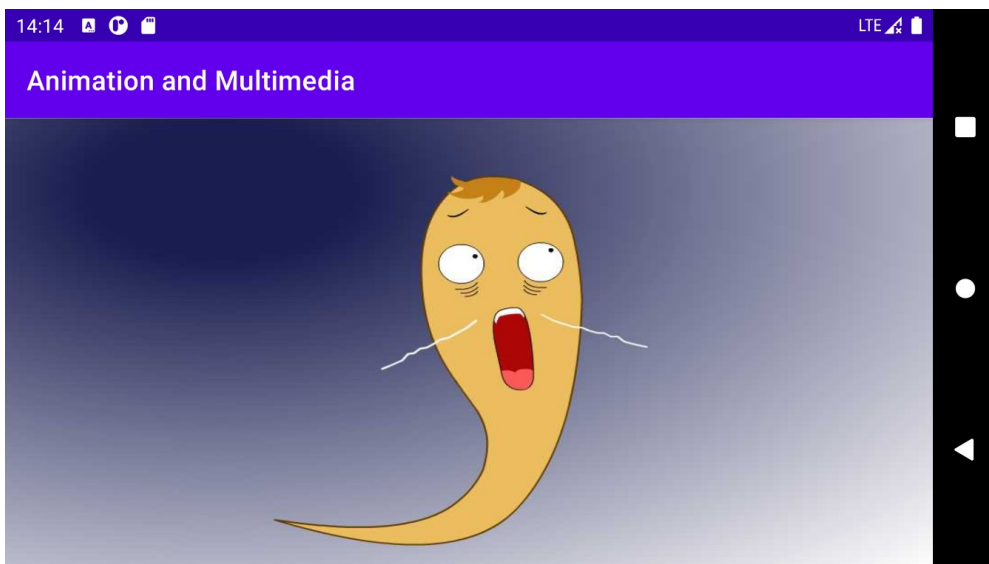


```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@drawable/img001" android:duration="60"/>
  <item android:drawable="@drawable/img002" android:duration="60"/>
  <item android:drawable="@drawable/img003" android:duration="60"/>
  <item android:drawable="@drawable/img004" android:duration="60"/>
  <item android:drawable="@drawable/img005" android:duration="60"/>
  <item android:drawable="@drawable/img006" android:duration="60"/>
</animation-list>
```

```
<LinearLayout
  android:id="@+id/linearLayout"
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:background="@drawable/fairy"
  tools:context="com.example.animationandmultimedia.MainActivity">
</LinearLayout>
```



Small Demo



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.animationandmultimedia">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.ServiceApplication">
        <activity android:name=".MainActivity" android:screenOrientation="landscape">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```
public class MainActivity extends AppCompatActivity {
    private boolean flag = true;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        LinearLayout linearLayout = (LinearLayout)findViewById(R.id.LinearLayout);
        final AnimationDrawable animationDrawable =
            (AnimationDrawable)linearLayout.getBackground();
        linearLayout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(flag){
                    animationDrawable.start();
                    flag = false;
                }else{
                    animationDrawable.stop();
                    flag = true;
                }
            }
        });
    }
}
```

Tween Animation

- **With the start and end key frames, system is responsible in computing and generating the intermediate frames**
 - **Android provides four types of tween animations**
 - **AlphaAnimation**
 - **RotateAnimation**
 - **ScaleAnimation**
 - **Translate Animation**

AlphaAnimation

■ Alternating alpha degree of components

□ Configuration

□ Start alpha degree, i.e., fromAlpha

□ End alpha degree, i.e., toAlpha

□ Duration

```
<set xmlns:android=http://schemas.android.com/apk/res/android
    android:interpolator="@[package:] anim/interpolator_resource">
    <alpha
        android:repeatMode="reverse|restart"
        android:repeatCount="specificTimes|infinite"
        android:duration="Integer"
        android:fromAlpha="float"
        android:toAlpha="float" />
</set>
```

@android:anim/linear_interpolator
@android:anim/accelerate_interpolator
@android:anim/decelerate_interpolator
@android:anim/accelerate_decelerate_interpolator
@android:anim/cycle_interpolator
@android:anim/bounce_interpolator
@android:anim/anticipate_overshoot_interpolator
@android:anim/overshoot_interpolator
@android:anim/anticipate_interpolator

android:interpolator	Speed of playing animation Linear speed, accelerate/decelerate, parabolic
android:repeatMode	Reverse, restart
android:repeatCount	Particular times, infinite
android:duration	Unit: millisecond

RotateAnimation

■ Alternating rotation angle

□ Configuration

□ Start rotation degree, i.e., fromDegree

□ End rotation degree, i.e., toDegree

□ Duration

```
<set xmlns:android=http://schemas.android.com/apk/res/android
    android:interpolator="@[package:] anim/interpolator_resource">
    <rotate
        android:fromDegrees="float"
        android:toDegrees="float"
        android:pivotX="float"
        android:pivotY="float"
        android:repeatMode="reverse|restart"
        android:repeatCount="specificTimes|infinite"
        android:duration="Integer" />
</set>
```

android:interpolator	Speed of playing animation Linear speed, accelerate/decelerate, parabolic
android:fromDegrees	Degree at beginning
android:toDegrees	Degree at end
android:pivotX	X coordinate of pivot
android:pivotY	Y coordinate of pivot
android:repeatMode	Reverse, restart
android:repeatCount	Particular times, infinite
android:duration	Unit: millisecond

ScaleAnimation

■ Alternating scaling coefficient

□ Configuration

□ Start scaling coefficient

□ End scaling coefficient

□ Duration

```
<set xmlns:android=http://schemas.android.com/apk/res/android
    android:interpolator="@[package:] anim/interpolator_resource">
    <scale
        android:fromXScale="float"
        android:toXScale="float"
        android:fromYScale="float"
        android:toYScale="float"
        android:pivotX="float"
        android:pivotY="float"
        android:repeatMode="reverse|restart"
        android:repeatCount="specificTimes|infinite"
        android:duration="Integer" />
</set>
```

android:interpolator	Speed of playing animation Linear speed, accelerate/decelerate, parabolic
android:fromXScale	Scaling coefficient on horizontal direction at beginning, scaling coefficient=1.0 means no scale
android:toXScale	Scaling coefficient on horizontal direction at end, scaling coefficient=1.0 means no scale
android:fromYScale	Scaling coefficient on vertical direction at beginning, scaling coefficient=1.0 means no scale
android:toYScale	Scaling coefficient on vertical direction at end, scaling coefficient=1.0 means no scale
android:pivotX	X coordinate of pivot
android:pivotY	Y coordinate of pivot
android:repeatMode	Reverse, restart
android:repeatCount	Particular times, infinite
android:duration	Unit: millisecond

Translate Animation

■ Alternating positions

□ Configuration

□ Start position

□ End position

□ Duration

```
<set xmlns:android=http://schemas.android.com/apk/res/android
    android:interpolator="@[package:] anim/interpolator_resource">
    <translate
        android:fromXDelta="float"
        android:toXDelta="float"
        android:fromYDelta="float"
        android:toYDelta="float"
        android:repeatMode="reverse|restart"
        android:repeatCount="specificTimes|infinite"
        android:duration="Integer" />
    </set>
```

android:interpolator	Speed of playing animation Linear speed, accelerate/decelerate, parabolic
android:fromXDelta	Position in horizontal direction at beginning
android:toXDelta	Position in horizontal direction at end
android:fromYDelta	Position in vertical direction at beginning
android:toYDelta	Position in vertical direction at end
android:repeatMode	Reverse, restart
android:repeatCount	Particular times, infinite
android:duration	Unit: millisecond

Small Demo



```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.tweenanimation.MainActivity">
    <ViewFlipper
        android:id="@+id/flipper"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </ViewFlipper>
</RelativeLayout>
```

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha android:fromAlpha="0"
        android:toAlpha="1"
        android:duration="4000"/>
</set>

<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha android:fromAlpha="1"
        android:toAlpha="0"
        android:duration="2000"/>
</set>
```

Small Demo

11:21

LTE

Tween Animation



```
public class MainActivity extends AppCompatActivity implements GestureDetector.OnGestureListener {
    ViewFlipper flipper;
    GestureDetector detector;
    Animation[] animations = new Animation[2];
    final int distance = 50;
    private int[] images = new int[]{R.drawable.img01,R.drawable.img02,R.drawable.img03,
        R.drawable.img04,R.drawable.img05,R.drawable.img06,R.drawable.img07,
        R.drawable.img08,R.drawable.img09};
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    detector = new GestureDetector(this, (GestureDetector.OnGestureListener) this);
```

```
    flipper = (ViewFlipper)findViewById(R.id.flipper);
```

```
    for(int i = 0; i < images.length; i++){
```

```
        ImageView imageView = new ImageView(this);
```

```
        imageView.setImageResource(images[i]);
```

```
        flipper.addView(imageView);
```

```
    }
```

```
    animations[0] = AnimationUtils.loadAnimation(this,R.anim.anim_alpha_in);
```

```
    animations[1] = AnimationUtils.loadAnimation(this, R.anim.anim_alpha_out);
```

```
    }
```

@Override

```
public boolean onDown(MotionEvent e) {
```

```
    return false;
```

```
}
```

@Override

```
public void onShowPress(MotionEvent e) {
```

```
}
```

@Override

```
public boolean onSingleTapUp(MotionEvent e) {
```

```
    return false;
```

```
}
```

@Override

```
public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {
```

```
    return false;
```

```
}
```

@Override

```
public void onLongPress(MotionEvent e) {
```

```
}
```

@Override

```
public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
```

```
    flipper.setInAnimation(animations[0]);
```

```
    flipper.setOutAnimation(animations[1]);
```

```
    if(e1.getX()-e2.getX() > distance){
```

```
        flipper.showPrevious();
```

```
        return true;
```

```
    }else if(e2.getX()-e1.getX() > distance){
```

```
        flipper.showNext();
```

```
        return true;
```

```
    }
```

```
    return false;
```

```
}
```

@Override

```
public boolean onTouchEvent(MotionEvent event) {
```

```
    return detector.onTouchEvent(event);
```

```
}
```

```
}
```

Play audio and video

■ Support to play all the usual formats

- Audio: .mp3, .3gp, .ogg, .wav, etc.
- Video: .3gp, .mp4, etc.

■ Play audio with MediaPlayer

- Create MediaPlayer class
- Load audio file resource
- Call start() or any other methods

<code>create(Context context, int resid)</code>	Create MediaPlayer object according to specific resource ID
<code>create(Context context, Uri uri)</code>	Create MediaPlayer object according to specific URI
<code>setDataSource()</code>	Specify resource to be loaded
<code>prepare()</code>	Prepare to play before playing audio
<code>start()</code>	Start to play
<code>stop()</code>	Stop playing
<code>pause()</code>	Pause to play
<code>reset()</code>	Reset MediaPlayer to initial status

To play resource from Internet, the app should be authorized with permission.
<uses-permission android:name="android.permission.INTERNET">

Small Demo



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    tools:context="com.example.musicplayer.MainActivity">
    <ImageButton
        android:id="@+id/btn_play"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="@dimen/margin_bottom"
        android:layout_marginLeft="@dimen/margin"
        android:background="@color/btn_Background"
        android:src="@drawable/play" />
    <ImageButton
        android:id="@+id/btn_stop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_marginBottom="@dimen/margin_bottom"
        android:layout_marginRight="@dimen/margin"
        android:background="@color/btn_Background"
        android:src="@drawable/stop" />
</RelativeLayout>
```


Small Demo



```
public class MainActivity extends Activity {
    private MediaPlayer player;
    private boolean isPause = false;
    private File file;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        final ImageButton btn_play = (ImageButton)findViewById(R.id.btn_play);
        final ImageButton btn_stop = (ImageButton)findViewById(R.id.btn_stop);
        player = MediaPlayer.create(MainActivity.this, com.example.musicplayer.R.raw.memoryLane);
        player.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
            @Override
            public void onCompletion(MediaPlayer mp) {
                play();
            }
        });
        btn_play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(player.isPlaying()&&!isPause){
                    player.pause();
                    isPause = true;
                    ((ImageButton)v).setImageDrawable(getResources().getDrawable(R.drawable.play, null));
                    isPause = false;
                }else{
                    player.start();
                    ((ImageButton)v).setImageDrawable(getResources().getDrawable(R.drawable.pause));
                    isPause = false;
                }
            }
        });
        btn_stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                player.stop();
                btn_play.setImageDrawable(getResources().getDrawable(R.drawable.play, null));
            }
        });
    }
}
```

Small Demo



```
@Override
protected void onDestroy() {
    if(player.isPlaying()){
        player.stop();
    }
    player.release();
    super.onDestroy();
}

private void play(){
    try{
        player.reset();
        player = MediaPlayer.create(this, com.example.musicplayer.R.raw.memoryLane);
        player.start();
    }catch(Exception e){
        e.printStackTrace();
    }
}
}
```


SoundPool playing audio

■ Simultaneously playing multiple small audios with less resource consumption

□ Steps

□ Create SoundPool object

`SoundPool(int maxStreams, int streamType, int srcQuality)`

- ✓ maxStreams: specifies the number of candidate audios
- ✓ streamType: specifies the audio type with AudioManager class, e.g., `STREAM_MUSIC`
- ✓ srcQuality: specifies the audio quality, default as 0

□ Load audio to play

`SoundPool soundpool = new SoundPool(10, AudioManager.STREAM_MUSIC, 0);`

□ `public int load(Context context, int resId, int priority)`

□ `public int load(String path, int priority)`

□ `public int load(AssetFileDescriptor afd, int priority)`

Higher priority, larger number

□ `public int load(FileDescriptor fd, long offset, long length, int priority)`

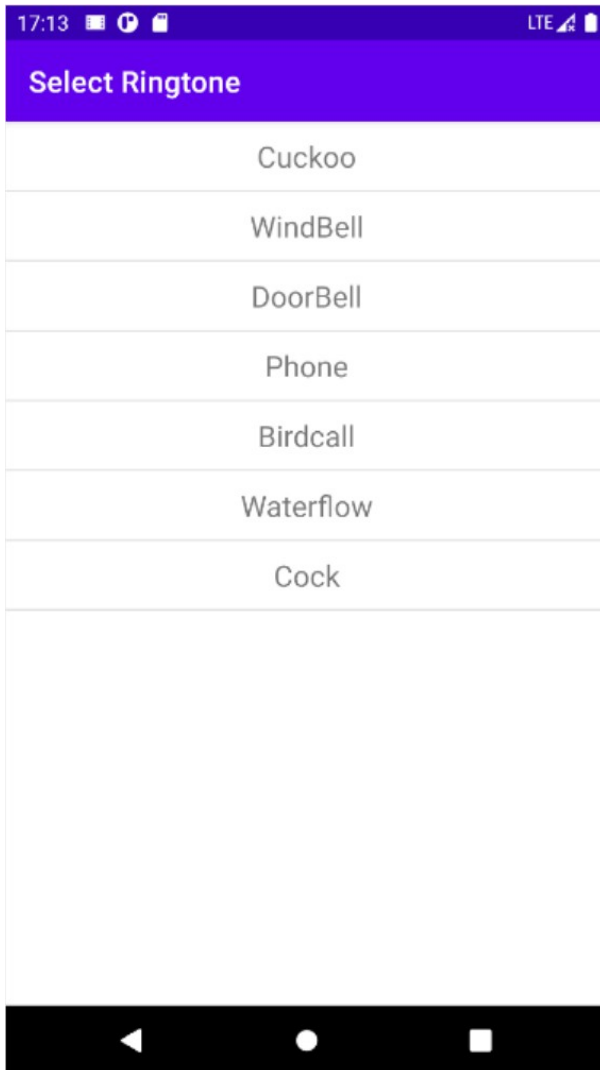
□ Play audio

□ `play(int soundID, float leftVolume, float rightVolume, int priority, int loop, float rate)`

loop: 0 = non-loop; -1 = loop

rate: 1 = normal; 0.5 = min, 2 = max

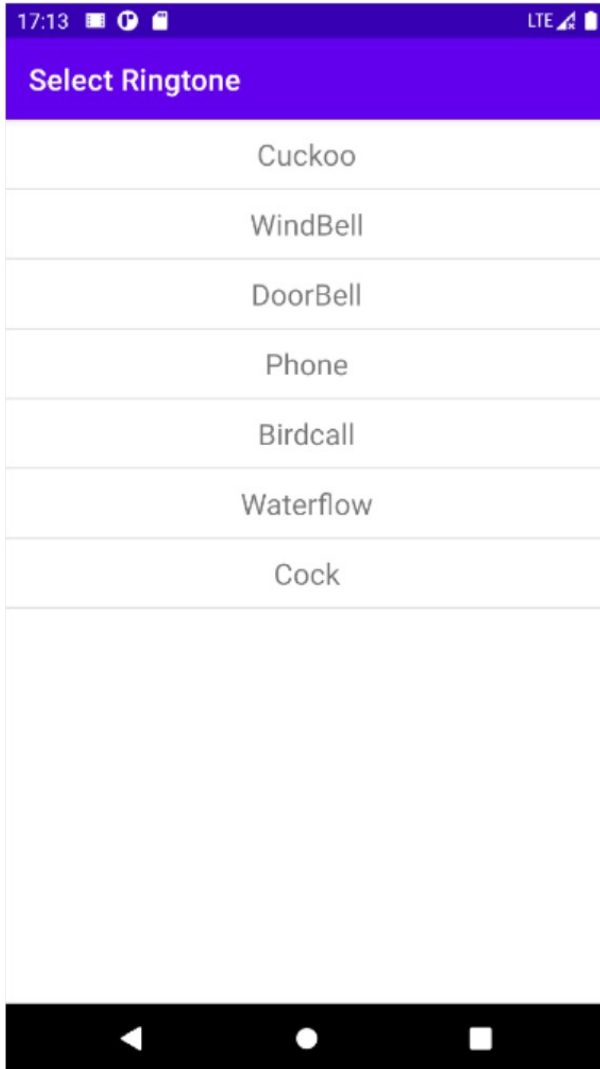
Small Demo



```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.selectringtone.MainActivity">
    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </ListView>
</RelativeLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:layout_gravity="center"
        android:id="@+id/title"
        android:textSize="@dimen/text_size"/>
</LinearLayout>
```

Small Demo



```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView listView = (ListView)findViewById(R.id.listView);
        String[] title = new String[]{"Cuckoo", "WindBell", "DoorBell", "Phone", "Birdcall", "Waterwave", "Cock"};
        List<Map<String, Object>> listItems = new ArrayList<Map<String, Object>>();
        for(int i = 0; i < title.length; i++){
            Map<String, Object> map = new HashMap<String, Object>();
            map.put("name", title[i]);
            listItems.add(map);
        }
        AudioAttributes audioAttributes = new AudioAttributes.Builder().setUsage(AudioAttributes.USAGE_GAME).
            setContentType(AudioAttributes.CONTENT_TYPE_MUSIC).build();
        final SoundPool soundPool = new SoundPool.Builder().setAudioAttributes(audioAttributes).setMaxStreams(10).build();
        final HashMap<Integer, Integer> soundmap = new HashMap<Integer, Integer>();
        soundmap.put(0, soundPool.load(this, R.raw.cuckoo, 1));
        soundmap.put(1, soundPool.load(this, R.raw.chimes, 1));
        soundmap.put(2, soundPool.load(this, R.raw.notify, 1));
        soundmap.put(3, soundPool.load(this, R.raw.ringout, 1));
        soundmap.put(4, soundPool.load(this, R.raw.bird, 1));
        soundmap.put(5, soundPool.load(this, R.raw.water, 1));
        soundmap.put(6, soundPool.load(this, R.raw.cock, 1));
        SimpleAdapter adapter = new SimpleAdapter(this, listItems,
            R.layout.main, new String[]{"name"}, new int[]{
                R.id.title});
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                Map<String, Object> map = (Map<String, Object>) parent.getItemAtPosition(position);
                soundPool.play(soundmap.get(position), 1, 1, 0, 0, 1);
            }
        });
    }
}
```

VideoView playing videos

■ Insert VideoView component in Layout, and call the method in Activity to play videos

□ setVideoURL() or setVideoPath()

```
<VideoView  
    attributeList>  
</VideoView>
```

android:id	ID of component
android:background	Image or pure color background
android:layout_gravity	Specifies the alignment direction
android:layout_width	Specifies the width of VideoView
android:layout_height	Specifies the height of VideoView

Control Camera to take photos

■ Using android.hardware.Camera class

- Camera class doesn't have constructor
- Configure Camera.Parameters to set the necessary factors for capturing
- Call startPreview() method to preview scene
- Call takePicture() method to capture sight
- Call stopPreview() method to complete preview
- Call release() method to complete using camera

<code>getParameters()</code>	Obtains the parameters of camera
<code>Camera.open()</code>	To open camera
<code>release()</code>	Completes using camera
<code>setParameters(Camera.Parameters parameters)</code>	Configures the parameters for capturing sights
<code>setPreviewDisplay(SurfaceHolder holder)</code>	Specifies a SurfaceView to display the preview window
<code>startPreview()</code>	Invokes preview windows
<code>takePicture(Camera.ShutterCallback shutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg)</code>	Takes a photo
<code>stopPreview()</code>	Stop camera

