



广东岭南职业技术学院

Android 移动开发基础案例教程

第 10 章 高级编程



- 图形图像处理
- 动画、多媒体
- Fragment
- Android5.0 新特性



作业点评

- GET 方式提交数据和 POST 方式提交数据的区别。
- Handler 消息机制中四个关键对象的作用。



预习检查

- Android 中的补间动画与逐帧动画的区别
- Fragment 的生命周期



主讲内容

10.1 图形图像处理

10.2 动画

10.3 多媒体

10.4 Fragment



图形图像处理

图形图像处理常用类

- Bitmap 是可以获取图像文件信息，进行图像剪切、旋转、缩放等操作，并可以指定格式保存图像文件。
- BitmapFactory 是位图工厂，它是一个工具类。
- Paint 代表画笔，用来描述图形的颜色及风格。
- Canvas 代表画布，通过该类提供的方法，可以绘制各种图形。
- Matrix 为图片添加特效的效果。



10.1.1 Bitmap 类

Bitmap 常用方法

`createBitmap(int width, int height, Config config)`

创建位图， `width` 代表要创建的图片的宽度， `height` 代表高度， `config` 代表图片的配置信息

`createBitmap(int colors[], int offset, int stride, int width, int height, Config config)`

使用颜色数组创建一个指定宽高的位图，颜色数组的个数为 `width*height`

`createBitmap(Bitmap src)`

使用源位图创建一个新的 Bitmap

`createBitmap(Bitmap source, int x, int y, int width, int height)`

从源位图的指定坐标开始“挖取”指定宽高的一块图像来创建新的 Bitmap 对象

`createBitmap(Bitmap source, int x, int y, int width, int height, Matrix m, boolean filter)`

从源位图的指定坐标开始“挖取”指定宽高的一块图像来创建新的 Bitmap 对象，并按照 Matrix 规则进行变换

`isRecycled()`

判断 Bitmap 对象是否被回收

`recycle()`

回收 Bitmap 对象

.....

.....



10.1.1 Bitmap 类

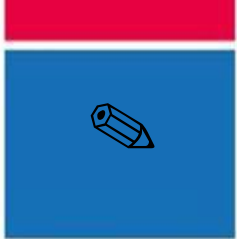


广东岭南职业技术学院
LINGNAN INSTITUTE OF
TECHNOLOGY

创建 Bitmap 头
文件

```
Bitmap.Config config = Config.ARGB_4444;
```

```
Bitmap bitmap = Bitmap.createBitmap( width, height, config );
```



10.1.2 BitmapFactory 类

BitmapFactory 常用方法

`decodeFile(String pathName)`

从指定文件中解析、创建 **Bitmap** 对象

`decodeStream(InputStream is)`

从指定输入流中解析、创建 **Bitmap** 对象

`decodeResource(Resources res, int id)`

根据给定的资源 id，从指定资源中解析、创建 **Bitmap** 对象

.....

.....

```
Bitmap bitmap = BitmapFactory.decodeFile("/sdcard/meinv.jpg");
```

```
Bitmap bitmap =
```

```
    BitmapFactory.decodeResource(getResources(), R.drawable.ic_launcher);
```



10.1.3 Paint 类

Paint 常用方法

Paint()

创建一个 Paint 对象，并使用默认属性

Paint(int flags)

创建一个 Paint 对象，并使用指定属性

setColor(int color)

设置颜色

setAlpha(int a)

设置透明度

setTextSize(float textSize)

设置绘制文本时的文字大小

setShadowLayer(float radius, float dx,
float dy, int color)

设置阴影，参数 radius 为阴影的角度；dx 和 dy 为阴影在 x 轴和 y 轴上的距离；color 为阴影的颜色

.....

.....

```
Paint paint = new Paint();
```

```
paint.setColor(Color.RED);
```



10.1.4 Canvas 类

Canvas 常用方法

`drawRect(Rect r, Paint paint)`

使用画笔画出指定矩形

`drawOval(RectF oval, Paint paint)`

使用画笔画出指定椭圆

`drawCircle(float cx, float cy, float
radius, Paint paint)`

使用画笔在指定位置画出指定半径的圆

`drawLine(float startX, float startY, float
stopX, float stopY, Paint paint)`

使用画笔在指定位置画线

`setTextSize(float textSize)`

设置绘制文本时的文字大小

`drawRoundRect(RectF rect, float rx,
float ry, Paint paint)`

使用画笔绘制指定圆角矩形，其中 `rx` 表示 X 轴圆角半径，`ry` 表示 Y 轴圆角半径

.....

.....



10.1.4 Canvas 类

```
protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
    Paint paint = new Paint();  
    paint.setColor(Color.RED);  
    Rect r = new Rect(40,40,200,100);  
    canvas.drawRect(r,paint);  
}
```



10.1.5 Matrix 类

Matrix 常用方法

Matrix()

创建一个唯一的 Matrix 对象

setRotate(float degrees)

将 Matrix 对象围绕 (0,0) 旋转 degrees 度

setRotate(float degrees, float px, float py)

将 Matrix 对象围绕指定位置 (px,py) 旋转 degrees 度

setScale(float sx, float sy)

对 Matrix 对象进行缩放，参数 sx 代表 X 轴上的缩放比例，sy 代表 Y 轴上的缩放比例

setSkew(float kx, float ky)

让 Matrix 对象倾斜，在 X 轴上倾斜 kx，在 Y 轴上倾斜 ky

setTranslate(float dx, float dy)

平移 Matrix 对象，(dx,dy) 为 Matrix 平移后的坐标

.....

.....

```
Matrix matrix = new Matrix();
```

```
matrix.setRotate(30);
```



10.1.6 实战演练——刮刮卡

1 功能描述： 触摸手机屏幕，使图片变透明

2 技术要点： 利用 Android 提供的类处理图片

- ① 用户交互界面的设计与实现
- ② XML 文件（ activity_main.xml ）的编写
- ③ 实体类（ MainActivity.java ）的编写
- ④ 界面逻辑代码的设计与实现

3 实现步骤：

案例代码（详见教材 Px—Pxx）





作业：完成实战演练 - 刮刮卡

要求：

- 1、参考老师讲解和教材中关于“实战演练 - 刮刮卡”实现的例子，完成“刮刮卡”的实现；
- 2、将项目中的 src 目录压缩好后，以“学号 + 姓名 01”命名发送到教师机上。



主讲内容

10.1 图形图像处理

10.2 动画

10.3 多媒体

10.4 Fragment



主讲内容

Android 中的动画

- 在 Android 开发中，避免不了用到动画，Android 中的动画分为补间动画和逐帧动画两种。

补间动画

逐帧动画



10.2.1 补间动画 (Tween Animation)

补间动画

- 在 Android 中，提供了四种补间动画：

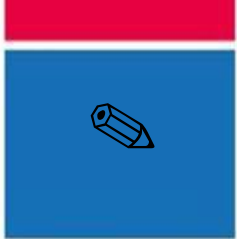




10.2.1 补间动画 (Tween Animation)

透明度渐变动画

- 透明度渐变动画是通过改变 View 组件透明度来实现的渐变效果。它主要通过为动画指定开始时的透明度、结束时的透明度以及动画持续时间来创建动画。



透明度渐变动画文件 (.xml)

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
```

```
  <alpha
```

```
    android:interpolator = "@android:anim/linear_interpolator"
```

```
    android:repeatMode = "reverse"
```

```
    android:repeatCount = "infinite"
```

```
    android:duration = "1000"
```

```
    android:fromAlpha = "1.0"
```

```
    android:toAlpha = "0.0"/>
```

```
</set>
```



10.2.1 补间动画 (Tween Animation)

旋转动画

- 旋转动画是通过为动画指定开始时的旋转角度、结束时的旋转角度以及动画播放时长来创建动画的。



旋转动画文件（.xml）

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
```

```
  <rotate
```

```
    android:fromDegrees="0"
```

```
    android:toDegrees="360"
```

```
    android:pivotX="50%"
```

```
    android:pivotY="50%"
```

```
    android:repeatMode="reverse"
```

```
    android:repeatCount="infinite"
```

```
    android:duration="1000"/>
```

```
</set>
```



10.2.1 补间动画 (Tween Animation)

缩放动画

- 缩放动画是通过为动画指定开始时的缩放系数、结束时的缩放系数以及动画持续时长来创建动画的。



缩放动画文件（.xml）

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
  <scale  
    android:repeatMode="reverse"  
    android:repeatCount="infinite"  
    android:duration="3000"  
    android:fromXScale="1.0"  
    android:fromYScale="1.0"  
    android:toXScale="0.5"  
    android:toYScale="0.5"  
    android:pivotX="50%"  
    android:pivotY="50%"/>  
</set>
```



10.2.1 补间动画 (Tween Animation)

平移动画

- 平移动画就是通过为动画指定开始位置、结束位置以及动画持续时长来创建动画的。



平移动画文件（.xml）

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
```

```
  <translate
```

```
    android:fromXDelta="0.0"
```

```
    android:fromYDelta="0.0"
```

```
    android:toXDelta="100"
```

```
    android:toYDelta="0.0"
```

```
    android:repeatCount="infinite"
```

```
    android:repeatMode="reverse"
```

```
    android:duration="4000"/>
```

```
</set>
```



10.2.1 补间动画 (Tween Animation)



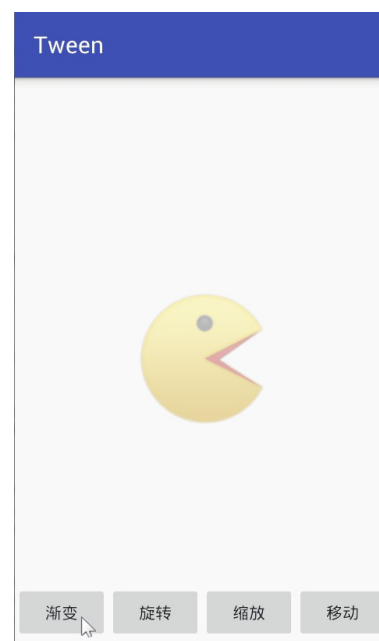
1 功能描述: 创建四种补间动画

2 技术要点: 编写四种动画文件

3 实现步骤:

- ① 用户交互界面的设计与实现
- ② 创建四种动画文件
- ③ 界面逻辑代码的设计与实现

案例代码 (详见教材 Px—Pxx)





作业：完成实战演练 - 补间动画

要求：

- 1、参考老师讲解和教材中关于“实战演练 - 补间动画”实现的例子，完成“补间动画”的实现；
- 2、将项目中的 app 目录压缩好后，以“学号 + 姓名”命名发送到教师机上。



10.2.2 逐帧动画 (Frame Animation)

逐帧动画

- 逐帧动画是按照准备好的静态图像顺序播放的，利用人眼的“视觉暂留”原理，造成动画的错觉。
- 逐帧动画的原理与放胶片看电影的原理是一样的，它们都是一张一张地播放事先准备好的静态图像。



10.2.2 逐帧动画 (Frame Animation)

1 功能描述: 播放逐帧动画

2 技术要点: 掌握实现逐帧动画步骤

① 用户交互界面的设计与实现

② 引入图片资源

③ 创建 Frame 动画文件

3 实现步骤: ④ 编写界面逻辑代码

案例代码 (详见教材 Px—Pxx)

Frame





作业：完成实战演练 - 逐帧动画

要求：

- 1、参考老师讲解和教材中关于“实战演练 - 逐帧动画”实现的例子，完成“逐帧动画”的实现；
- 2、将项目中的 src 目录压缩好后，以“学号 + 姓名 03”命名发送到教师机上。



主讲内容

10.1 图形图像处理

10.2 动画

10.3 多媒体

10.4 Fragment



10.3.1 MediaPlayer 播放音频

- MediaPlayer 是用于播放音频和视频的，它支持多种格式音频文件（3gp、mp4），并提供了非常全面的控制方法。

setDataSource()

设置要播放的音频文件的位置

prepare()

在开始播放之前调用这个方法完成准备工作

start()

开始或继续播放音频

pause()

暂停播放音频

reset()

将 MediaPlayer 对象重置到刚刚创建的状态

seekTo()

从指定位置开始播放音频

stop()

停止播放音频，调用该方法后 MediaPlayer 对象无法在播放音频

release()

释放掉与 MediaPlayer 对象相关的资源

isPlaying()

判断当前 MediaPlayer 是否正在播放音频

getDuration

获取载入的音频文件的时长

.....

.....



10.3.1 MediaPlayer 播放音频

创建 MediaPlayer

```
MediaPlayer mediaPlayer = new MediaPlayer();  
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
```

- MediaPlayer 接收音频类型有很多，其中有四种较为常用的类型：
 - AudioManager.STREAM_MUSIC （音乐）
 - AudioManager.STREAM_RING （响铃）
 - AudioManager.STREAM_ALARM （闹钟）
 - AudioManager.STREAM_NOTIFICATION （提示音）
- 音频类型不同占据的内存空间也不同，音频时间越短占的内存越小



10.3.1 MediaPlayer 播放音频

设置数据源

- 设置数据源有三种方式，分别是播放应用自带的音频文件、播放 SD 卡中的音频文件、播放网络音频文件。

```
mediaPlayer.create(this,R.raw.xxx);
```

```
mediaPlayer.setDataSource("mnt/sdcard/xxx.mp3");
```

```
mediaPlayer.setDataSource("http://www.xxx.mp3");
```



10.3.1 MediaPlayer 播放音频

播放本地音频文件

```
mediaPlayer.prepare();  
mediaPlayer.start();
```



10.3.1 MediaPlayer 播放音频

播放网络音频文件

```
mediaPlayer.prepareAsync();
mediaPlayer.setOnPreparedListener(new OnPreparedListener){
    public void onPrepared(MediaPlayer player){
        mediaPlayer.start();
    }
}
```



10.3.1 MediaPlayer 播放音频

暂停播放

```
if(mediaPlayer!=null && mediaPlayer.isPlaying()){  
    mediaPlayer.pause();  
}
```



10.3.1 MediaPlayer 播放音频

重新播放

- 播放状态下进行重播

```
if(mediaPlayer!=null && mediaPlayer.isPlaying()){  
    mediaPlayer.seekTo(0);  
    return;  
}
```

- 暂停状态下进行重播

```
if(mediaPlayer!=null){  
    mediaPlayer.seekTo(0);  
    mediaPlayer.start();  
}
```



10.3.1 MediaPlayer 播放音频

停止播放

```
if(mediaPlayer!=null && mediaPlayer.isPlaying()){  
    mediaPlayer.stop();  
    mediaPlayer.release();  
    mediaPlayer = null;  
}
```



10.3.2 VideoView 播放视频

VideoView

- VideoView 组件播放视频最简单，它将视频的显示和控制集于一身，因此，借助它就可以完成一个简易的视频播放器。

setVideoPath()

设置要播放的视频文件的位置

start()

开始或继续播放音频

pause()

暂停播放音频

resume()

将视频重新开始播放

seekTo()

从指定位置开始播放视频

isPlaying()

判断当前是否正在播放视频

getDuration()

获取载入的视频文件的时长

.....

.....



10.3.2 VideoView 播放视频

使用 VideoView

– 创建 VideoView

```
<VideoView  
    android:id="@+id/videoview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

– 视频的播放

```
VideoView videoView = (VideoView) findViewById(R.id.videoview);  
videoView.setVideoPath("mnt/sdcard/apple.avi");  
videoView.setVideoURI(Uri.parse("http://www.xxx.avi"));  
videoView.start();
```



10.3.2 VideoView 播放视频

添加控制器

- 控制器 `MediaController` 是一个包含媒体播放器（`MediaPlayer`）控件的视图，包含了一些典型的按钮如：播放 / 暂停（`Play/ Pause`）、倒带（`Rewind`）、快进（`Fast Forward`）与进度滑动器（`progress slider`）。它管理媒体播放器（`MediaController`）的状态以保持控件的同步。

```
MediaController controller = new MediaController(context);  
videoView.setMediaController(controller);
```



10.3.3 实战演练——视频播放器

1 功能描述: 播放 SD 卡中的视频文件

2 技术要点: 通过 VideoView 实现播放视频

3 实现步骤:

- ① 用户交互界面的设计与实现
- ② 界面逻辑代码的设计与实现

案例代码（详见教材 P_x—P_{xx}）





作业：完成实战演练 - 视频播放器

要求：

- 1、参考教材中关于“实战演练 - 视频播放器”实现的例子，完成“视频播放器”的实现；
- 2、将项目中的 src 目录压缩好后，以“学号 + 姓名 01”命名发送到教师机上。



主讲内容

10.1 图形图像处理

10.2 动画

10.3 多媒体

10.4 Fragment



10.4.1 Fragment 简介

Fragment 简介

- Fragment（碎片）是一种可以嵌入在 Activity 中的 UI 片段，与 Activity 非常相似，不仅包含布局，同时也具有自己的生命周期。



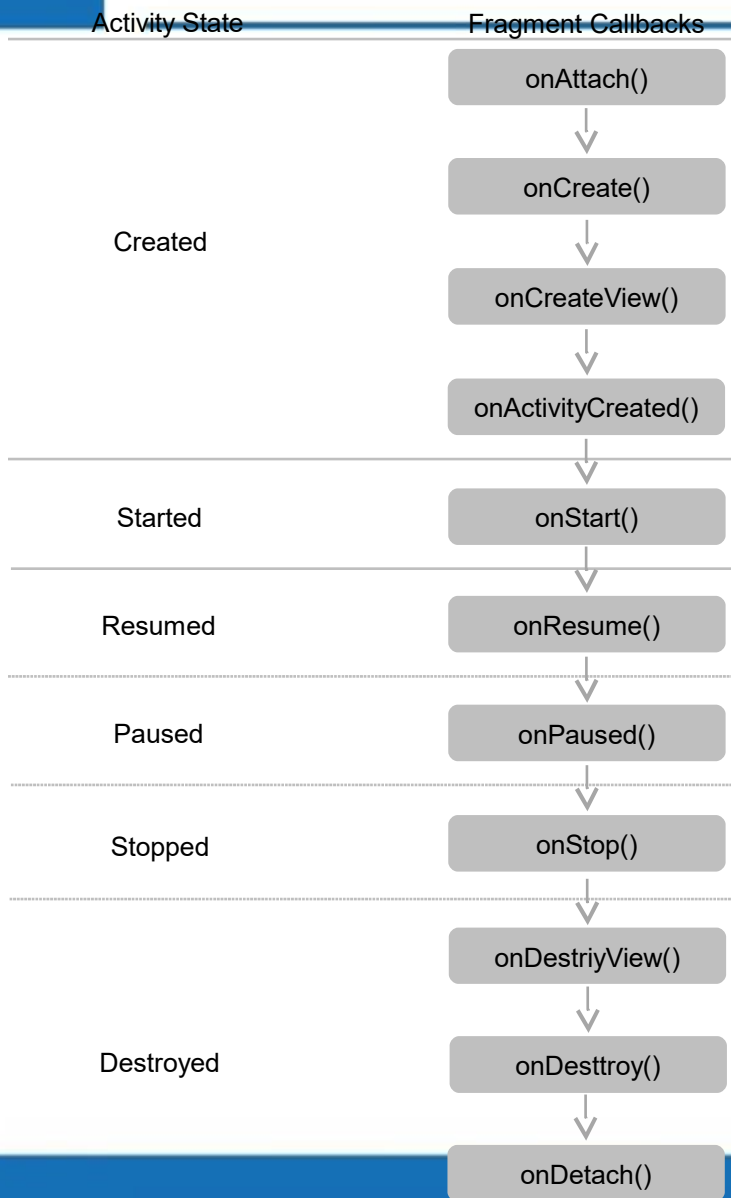
10.4.2 Fragment 的生命周期

生命周期

- Fragment 不能独立存在必须嵌入到 Activity 中使用，所以 Fragment 生命周期直接受所在的 Activity 影响。
- 当 Activity 暂停时，它拥有的所有 Fragment 都暂停，当 Activity 销毁时，它拥有的所有 Fragment 都被销毁。



10.4.2 Fragment 的生命周期

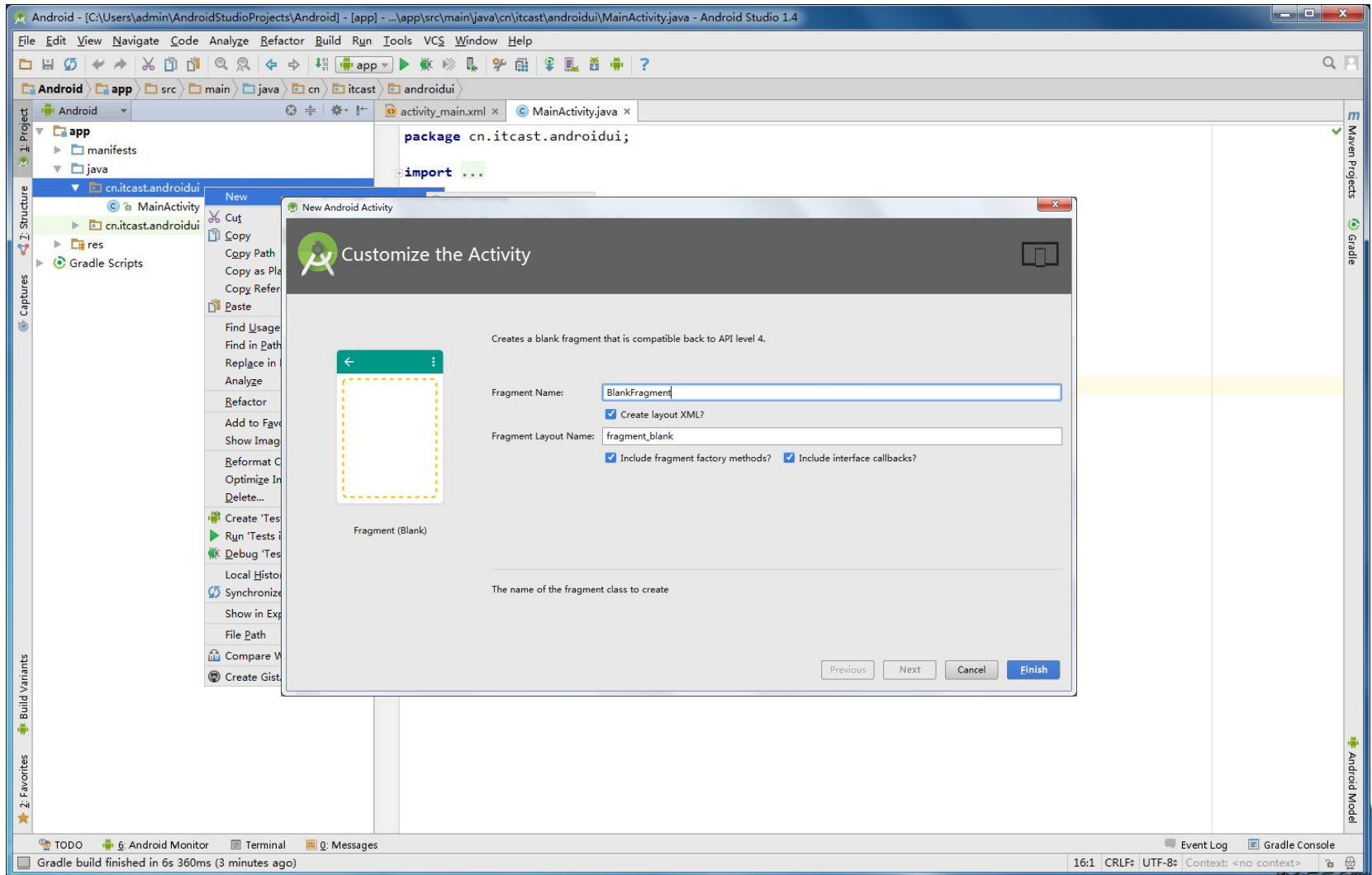




10.4.3 Fragment 的创建



广东岭南职业技术学院
LINGNAN INSTITUTE OF
TECHNOLOGY





10.4.3 Fragment 的创建

```
public class MyFragment extends Fragment {  
    public MyFragment() {  
        // Required empty public constructor  
    }  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.fragment_my, container, false);  
    }  
}
```



10.4.3 Fragment 的创建

- 在 Activity 中使用 Fragment 时，可以通过两种方式将 Fragment 添加到 Activity 中，一种是通过布局文件添加，一种是通过代码动态添加。
- 通过布局文件添加 Fragment

```
<fragment
```

```
    android:id="@+id/fragment"
```

```
    android:name="cn.itcast.myfragment.MyFragment"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">
```

```
</fragment>
```



10.4.3 Fragment 的创建

- 通过代码动态添加 Fragment

```
MyFragment fragment = new MyFragment();  
FragmentManager fm = getFragmentManager();  
FragmentTransaction beginTransaction = fm.beginTransaction();  
beginTransaction.add(R.id.rl,fragment);  
beginTransaction.commit();
```



10.4.4 实战演练——滑动切换界面

1 功能描述： 实现三个界面进行滑动切换

2 技术要点： 创建 Fragment 实现滑动效果

3 实现步骤：

- ① 用户交互界面的设计与实现
- ② 创建三个 Fragment
- ③ 界面逻辑代码的设计与实现



案例代码（详见教材 Px—Pxx）



作业：完成实战演练 - 滑动切换界面

要求：

- 1、参考教材中关于“实战演练 - 滑动切换界面”实现的例子，完成“滑动切换界面”的实现；
- 2、将项目中的 src 目录压缩好后，以“学号 + 姓名 02”命名发送到教师机上。



主讲内容



广东岭南职业技术学院
LINGNAN INSTITUTE OF
TECHNOLOGY

10.5 Android5.0 新特性



10.5.1 抽屉动画

抽屉动画

- Android5.0 中新增抽屉动画（ DrawerLayout ），通过抽屉动画可以实现侧滑效果。

```
<android.support.v4.widget.DrawerLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
....
```

```
....
```

```
....
```

```
</android.support.v4.widget.DrawerLayout>
```



10.5.1 抽屉动画

抽屉动画

- DrawerLayout 控件只需在布局文件中引入，不需要编写用户交互代码便可实现侧滑效果。



10.5.2 RecyclerView 控件

RecyclerView 控件

- Android5.0 之后，谷歌公司提供了一个用于在有限的窗口范围内显示大量数据的 RecyclerView 控件。RecyclerView 本身不参与任何视图相关的问题，它只负责回收和重用的工作。

```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/id_recyclerview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
</android.support.v7.widget.RecyclerView>
```



10.5.3 SwipeRefreshLayout 控件



SwipeRefreshLayout 控件

- 下拉刷新是一个使用非常广泛的功能，在 Android5.0 之后，谷歌公司推出了一个全新的控件 `SwipeRefreshLayout` 用于实现下拉刷新，使在开发过程中更加方便简洁。



10.5.3 SwipeRefreshLayout 控件

```
<android.support.v4.widget.SwipeRefreshLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <ScrollView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
        <TextView  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text=" 下拉刷新 " />  
        </ScrollView>  
    </android.support.v4.widget.SwipeRefreshLayout>
```



10.6 本章小结

本章详细讲解了图形图像处理、动画、多媒体以及 Fragment 等知识点。这些知识属于 Android 中的高级部分，因此要求初学者在学习本章之前，必须先熟练掌握前面讲解的知识，打好 Android 基础。



本章作业

- 请简要说明 Android 中有几种动画，以及每种动画的特点。
- 请简要说明 Fragment 生命周期包含哪些方法。



Thank You!

yx.boxuegu.com

