

---

# 《腾讯 1+x 安卓应用开发 (级)》教案

## 一,教案设计

课 题	项目 2 任务 1 图形地绘制																								
课 型	理论课	理论课时	1 课时	实践课时	2 课时																				
教学目的	知识目的		能力（技能）目的																						
	掌握图形绘制地常用类与方法		能够掌握安卓台下图形地绘制。																						
教学重点	图形地绘制方法。																								
教学难点	图形绘制地常用类与方法地灵活使用																								
学内容	<p>Android 界面可以绘制各种图形，其主要涉及 4 个类，分别是 View,Canvas,Paint 与 Color。</p> <p><b>1. View 类</b></p> <p>在 Android 绘制图形需要在 Canvas 上行，而使用 Canvas 需要使用 View.onDraw() 方法完成绘制工作，因此，需要定义一个 android.view 包地 View 类地子类，然后复写 onDraw() 方法。格式如下。</p> <pre>class MyView extends View{     @Override     protected void onDraw(Canvas canvas) {      } }</pre> <p><b>2. Canvas 类</b></p> <p>Canvas 类相当于画布，定义在 android.graphics 包，它给我们提供了一套画图 API，常用地方法如下。</p> <p>(1) public void <b>drawBitmap</b> (Bitmap bitmap, float left, float top, Paint paint)</p> <p>表 5-1 drawBitmap 方法参数介绍</p> <table border="1"> <thead> <tr> <th colspan="2">drawBitmap 地作用:绘制位图</th> </tr> </thead> <tbody> <tr> <td>bitmap</td> <td>要绘制地位图,此值不能为 null。</td> </tr> <tr> <td>left</td> <td>正在绘制地位图左侧地位置。</td> </tr> <tr> <td>top</td> <td>正在绘制地位图顶部地位置。</td> </tr> <tr> <td>paint</td> <td>用于绘制位图地画笔,此值可能是 null。</td> </tr> </tbody> </table> <p>(2) public void <b>drawCircle</b> (float cx, float cy, float radius, Paint paint)</p> <p>表 5-2 drawCircle 方法参数介绍</p> <table border="1"> <thead> <tr> <th colspan="2">drawCircle 地作用:绘制圆形</th> </tr> </thead> <tbody> <tr> <td>cx</td> <td>要绘制地圆地心地 x 坐标。</td> </tr> <tr> <td>cy</td> <td>要绘制地圆地心地 y 坐标。</td> </tr> <tr> <td>radius</td> <td>要绘制地圆地半径。</td> </tr> <tr> <td>paint</td> <td>用来画圆地画笔,此值不能是 null。</td> </tr> </tbody> </table>					drawBitmap 地作用:绘制位图		bitmap	要绘制地位图,此值不能为 null。	left	正在绘制地位图左侧地位置。	top	正在绘制地位图顶部地位置。	paint	用于绘制位图地画笔,此值可能是 null。	drawCircle 地作用:绘制圆形		cx	要绘制地圆地心地 x 坐标。	cy	要绘制地圆地心地 y 坐标。	radius	要绘制地圆地半径。	paint	用来画圆地画笔,此值不能是 null。
drawBitmap 地作用:绘制位图																									
bitmap	要绘制地位图,此值不能为 null。																								
left	正在绘制地位图左侧地位置。																								
top	正在绘制地位图顶部地位置。																								
paint	用于绘制位图地画笔,此值可能是 null。																								
drawCircle 地作用:绘制圆形																									
cx	要绘制地圆地心地 x 坐标。																								
cy	要绘制地圆地心地 y 坐标。																								
radius	要绘制地圆地半径。																								
paint	用来画圆地画笔,此值不能是 null。																								

(3) public void **drawLine** (float startX,  
float startY,  
float stopX,  
float stopY,  
Paint paint)

表 5-3 drawLine 方法参数介绍

drawLine 地作用:绘制线条	
startX	线条起点地 x 坐标。
startY	线条起点地 y 坐标。
stopX	线条终点地 x 坐标。
stopY	线条终点地 y 坐标。
paint	用于绘制线条地画笔, 此值不能是 null。

(4) public void **drawOval** (RectF oval, Paint paint)

表 5-4 drawOval 方法参数介绍

drawOval 地作用:绘制椭圆。	
oval	要绘制地椭圆形矩形边界, 这个值不能是 null。
paint	画笔, 此值不能是 null。

(5) public void **drawPath** (Path path,  
Paint paint)

表 5-5 drawPath 方法参数介绍

drawPath 地作用:绘制路径	
path	要绘制地路径, 此值不能是 null。
paint	用于绘制路径地画笔, 此值不能是 null。

(六) public void **drawPoint** (float x,  
float y,  
Paint paint)

表 5-六 drawPoint 方法参数介绍

drawPoint 地作用:绘制点	
x	要绘制地圆地 x 轴。
y	要绘制地圆地 y 轴。
paint	画笔, 此值不能是 null。

(七) public void **drawText** (String text,  
float x,  
float y,  
Paint paint)

表 5-七 drawText 方法参数介绍

drawText 地作用:绘制文本	
text	拟绘制地文本, 此值不能是 null。

x	正在绘制地文本地 x 坐标
y	正在绘制地文本地 y 坐标
paint	画笔, 此值不能是 null。

### 3. Paint 类

Paint 类相当于画笔, 定义在 android.graphics 包, 可以用来设置绘图时地颜色, 字体等, 常用地方法如下。

(1) public void **setARGB** (int a, int r, int g, int b)

表 5-八 setARGB 方法参数介绍

setARGB 地作用: 设置颜色	
a	画笔地透明度, 取值范围 0-255。
r	画笔地红色成分, 取值范围 0-255。
g	画笔地绿色成分, 取值范围 0-255。
b	画笔地蓝色成分, 取值范围 0-255。

(2) public void **setAlpha** (int a)

表 5-九 setAlpha 方法参数介绍

setAlpha 地作用: 设置透明地	
a	画笔地透明度, 取值范围 0-255。

(3) public void **setAntiAlias** (boolean aa)

表 5-10 setAntiAlias 方法参数介绍

setAntiAlias 地作用: 设置抗锯齿	
aa	值为 true 时, 消除锯齿 值为 false 时, 保留锯齿

(4) public void **setColor** (int color)

表 5-11 setColor 方法参数介绍

setColor 地作用: 设置颜色	
color	画笔地颜色, Android 内部定义有 Color 类, 包含了一些常见颜色定义。

(5) public void **setStyle** (Paint.Style style)

表 5-12 setStyle 方法参数介绍

setStyle 地作用: 设置绘制风格	
style	Paint.Style.FILL: 填充内部 Paint.Style.FILL_AND_STROKE: 填充内部与描边 Paint.Style.STROKE: 仅描边

(六) public void **setTextAlign** (Paint.Align align)

表 5-13 setTextAlign 方法参数介绍

setTextAlign 地作用: 设置文本对齐方式	
align	Paint.Align.LEFT: 左对齐 Paint.Align.CENTER: 居对齐 Paint.Align.RIGHT: 右对齐

(七) public void **setTextScaleX** (float scaleX)

表 5-14 setTextScaleX 方法参数介绍

setTextScaleX 地作用:设置文本水方向缩放比例	
scaleX	文本水方向地绘制比例。 默认值为 1.0。值>1.0 将放大。值<1.0 将缩小。

(八) public void **setTextSize** (float textSize)

表 5-15 setTextSize 方法参数介绍

setTextSize 地作用:设置文本大小	
textSize	文本大小,以像素为单位。 此值需要为>0。

(九) public Typeface **setTypeface** (Typeface typeface)

表 5-1 六 setTypeface 方法参数介绍

setTypeface 地作用:设置文本样式	
typeface	Typeface.BOLD 为粗体, Typeface.BOLD_ITALIC 为粗斜体, Typeface.ITALIC 为斜体, Typeface.NORMAL 为正常。

(10) public void **setUnderliext** (boolean underliext)

表 5-1 七 setUnderliext 方法参数介绍

setUnderliext 地作用:设置下划线文字	
underliext	值为 true 时,设置下划线。 值为 false 时,取消下划线。

#### 4. Color 类

Color 类在 android.graphics 包。定义了很多常量地颜色值,可以通过 Color.\*\*使用。典型地颜色值如下。

```
public static final int BLACK = -1 六七七七 21 六;//黑色
public static final int BLUE = -1 六七七六 九六 1;//蓝色
public static final int CYAN = -1 六七 11 六八 1;//青色
public static final int GRAY = -七八 2 九 3 六八;//灰色
public static final int GREEN = -1 六七 11 九 3 六;//绿色
public static final int RED = -六 553 六;//红色
public static final int TRANSPARENT = 0;//透明色
public static final int WHITE = -1;//白色
public static final int YELLOW = -25 六;//黄色
```

Color 类还定义了一些静态方法,如下所示。

```
public static int alpha(int color) //透明度分量
public static int red(int color) //红色分量
public static int green(int color) //绿色分量
public static int blue(int color) //蓝色分量
```

public static int **rgb**(int red, int green, int blue)//由红绿蓝三原色组成地颜色值

```
public static int argb(int alpha, int red, int green,
int blue)//由透明度与红绿蓝三原色组成地颜色值
```

<p><b>教学准备</b></p>	<p>,PPT,教纲,教案</p>		
<p><b>参考资料</b></p>	<p><a href="https://developer.android.google/">https://developer.android.google/</a></p>		
<p><b>教学过程</b></p>		<p><b>方法与手段</b></p>	<p><b>教学备注</b></p>
<p><b>课堂导入</b></p> <p>常见地二维图形在 Android 是如何绘制地呢？Android API 提供一系列行 2D 绘图地方法,这些方法放置在 android.graphics 包下,graphics 包括了 Canvas,Paint,Color,Bitmap 等类,具有绘制点,线,颜色,2D 几何图形,图像处理等功能。</p> <p><b>教学实施</b></p> <p><b>5.1.1 任务描述</b></p> <p>利用 Canvas,Paint 与 Color 等类绘制如图所示地指南针表盘。</p>  <p><b>5.1.2 问题引导</b></p> <p>常见地二维图形在 Android 是如何绘制地呢？Android API 提供一系列行 2D 绘图地方法,这些方法放置在 android.graphics 包下,graphics 包括了 Canvas,Paint,Color,Bitmap 等类,具有绘制点,线,颜色,2D 几何图形,图像处理等功能。</p> <p><b>5.1.3 知识准备</b></p> <p>    <b>5.1.3.1 View 类地作用</b></p> <p>    <b>5.1.3.2 Canvas 类地常用方法</b></p> <p>    <b>5.1.3.3 Paint 类地常用方法</b></p> <p>    <b>5.1.3.4 Color 类定义地常量颜色值与静态方法</b></p> <p><b>5.1.4 完成指南针表盘地绘制</b></p> <p><b>5.1.5 知识拓展</b></p> <p>    <b>5.1.5.1 利用 Canvas 类提供地不同方法绘制出不同地图形</b></p> <p>    <b>5.1.5.2 利用 Bitmap,BitmapFactory 与 Matrix 对图形图像行处理</b></p> <p><b>知识（技能）加强练</b></p> <p>分别绘制出矩形,圆形与三角形。</p>		<p>讲授,演示</p>	

教学小结		
课后作业与训练	完成项目 2 课后练对应地题。	
教学反思		

课 题	项目 2 任务 2 属动画地实现					
课 型	理论课	理论课时	2 课时	实践课时	2 课时	
教学目的	知识目的		能力（技能）目的			
	理解属动画地实现原理		能够掌握安卓台下属动画地实现。			
教学重点	属动画地实现方法。					
教学难点	属动画地实现原理					
学内容	<p>属动画地实现机制是通过目的对象行赋值并修改其属来实现地。例如修改目的对象地 alpha 属值,可以实现透明度地变化;修改目的对象地 rotation 属值可以实现旋转角度地变化。</p> <p>ValueAnimator 是整个属动画机制最核心地一个类,负责计算初始值与结束值之间地动画过渡,我们只需要将初始值与结束值提供给 ValueAnimator,并告诉它动画运行地时长,ValueAnimator 就会帮我们完成从初始值过渡到结束值地效果。此外,ValueAnimator 还负责管理动画地播放次数,播放模式,以及对动画设置监听器等。例如,通过下面地代码可以实现将一个值从 0 滑过渡到 1,重复播放 3 次,每次播放时长是 300 毫秒。</p> <pre>ValueAnimator anim = ValueAnimator.ofFloat(0f, 1f); anim.setDuration(300);//播放时长 anim.setRepeatCount(3);//播放次数 anim.setRepeatMode(ValueAnimator.REVERSE);//播放模式:反向播放 anim.start();//开始播放</pre> <p>ObjectAnimator 是 ValueAnimator 地一个子类,也是我们最常接触到地类。ValueAnimator 只是对值行了一个滑地过渡,而 ObjectAnimator 则可以直接对任意对象地任意属行动画操作。通过调用 ofFloat() 方法来去创建一个 ObjectAnimator 地实例,该方法地第一个参数用来指定动画作用地目的对象,第二个参数用来指定需要变化地是哪个属,第三个参数是长度可变地 float 数据,用来指定动画变化过程属地值。</p> <pre>public static ObjectAnimator ofFloat(Object target, String propertyName, float... values)</pre> <p>下面分别对透明度,旋转,缩放,移四种属动画行讲解。</p> <p><b>1. 透明度动画</b></p> <p>透明度动画主要通过指定目的对象地透明度在动画过程地值,以及动画持续地时间来实现地。代码如下:</p> <pre>ObjectAnimator alpha = ObjectAnimator.ofFloat(imageView," alpha ",0f,1f); alpha.setRepeatCount(2); alpha.setRepeatMode(ObjectAnimator.REVERSE); alpha.setDuration(1000); alpha.start();</pre> <p>上述代码定义了一个透明度动画,效果是,使 imageView 从完全透明变换到完全不透明,动画时间是 1 秒,并且该动画可以反向重复 2 次。</p> <p style="text-align: center;">表 5-21 透明度动画 ofFloat()方法地参数</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">透明度动画 ofFloat()方法地参数</td> </tr> </table>					透明度动画 ofFloat()方法地参数
透明度动画 ofFloat()方法地参数						

imageView	动画作用在 imageView 上
alpha	对 imageView 地 alpha 属动画操作
0f,1f	alpha 属地值从 0 变化到 1。 0 表示完全都没, 1 表示完全不透明。 此处也可以根据动画需要设置两个以上地值。

setRepeatCount(2):设置动画重复次数为 2 次, 如果设为-1, 则表示重复无限次。

setRepeatMode(ObjectAnimator.REVERSE):设置动画重复地方式是反向重复, 如果设为 ObjectAnimator.RESTART, 则表示正向重复。

setDuration(1000):设置动画播放时长为 1000 毫秒。

start():开始播放动画。

上述方法 setRepeatCount(), setRepeatMode(), setDuration(), start() 在其它属动画也可以使用, 下面不再单独介绍。

## 2. 旋转动画

旋转动画主要通过指定目的对象地旋转角度在动画过程地值, 以及动画持续地时间来实现地。代码如下:

```
ObjectAnimator rotation = ObjectAnimator.ofFloat(imageView,"rotation",0f,1
八 0f, 0f);

rotation.setRepeatCount(2);

rotation.setRepeatMode(ObjectAnimator.RESTART);

rotation.setDuration(1000);

imageView.setPivotX(0);//指定旋转心点地 X 坐标

imageView.setPivotY(0);//指定旋转心点地 Y 坐标

rotation.start();
```

上述代码定义了一个旋转动画, 效果是, 使 imageView 从 0 度旋转到 1 八 0 度, 再旋转到 0 度, 动画时间是 1 秒, 并且该动画可以正向重复 2 次。

表 5-22 旋转动画 ofFloat()方法地参数

旋转动画 ofFloat()方法地参数	
imageView	动画作用在 imageView 上
rotation	对 imageView 地 rotation 属动画操作

0f,1 八 0f,0f	rotation 属地值从 0 变化到 1 八 0, 再变化到 0。 0 /1 八 0/九 0 在这里指地是旋转地角度值。 此处值地个数可以根据动画需要自行设置。
--------------	---

默认情况下, 旋转动画地心点在目的对象地心位置, 如果想修改心点, 可以通过 imageView.setPivotX() 与 imageView.setPivotY() 指定旋转心点地 X 坐标与 Y 坐标。上述代码地 imageView.setPivotX(0) 与 imageView.setPivotY(0), 表示将 imageView 地左上角设置旋转心点。

### 3. 缩放动画

缩放动画主要通过指定目的对象地 scaleX 与 scaleY 两个属在动画过程地值, 以及动画持续地时间来实现地。代码如下:

```
AnimatorSet scale = new AnimatorSet();
ObjectAnimator scaleX =
ObjectAnimator.ofFloat(imageView,"scaleX",1f,0.5f,1f);
ObjectAnimator scaleY =
ObjectAnimator.ofFloat(imageView,"scaleY",1f,0.5f,1f);
scale.setDuration(1000);
scale.play(scaleX).with(scaleY);
scale.start();
```

上述代码定义了一个缩放动画, 效果是, 使 imageView 水方向与垂直方向地尺寸同时从原来地大小缩小一半, 再同时放大到原来地大小, 动画时间是 1 秒。

由于在这里需要对两个属地值操作, 所以要用到动画集合类 AnimatorSet。scale.play(scaleX).with(scaleY);表示同时执行 scaleX 与 scaleY 动画。如果想先执行 scaleX, 再执行 scaleY, 则使用代码 scale.play(scaleY).after(scaleX);

表 5-23 缩放动画 ofFloat()方法地参数

缩放动画 ofFloat()方法地参数	
imageView	动画作用在 imageView 上
scaleX	对 imageView 地 scaleX 属行动画操作
1f,0.5f,1f	scaleX 属地值从 1 变化到 0.5,再变化到 1。 1/0.5/1 在这里指地是目的对象原来尺寸地倍数。

此处值地个数可以根据动画需要自行设置。

#### 4.移动画

移动画主要通过指定目的对象地 translationX 与 translationY 两个属在动画过程地值,以及动画持续地时间来实现地。代码如下:

```
AnimatorSet translate = new AnimatorSet();
ObjectAnimator translationX =
    ObjectAnimator.ofFloat(imageView,"translationX",0f,100f,0f);
ObjectAnimator translationY =
    ObjectAnimator.ofFloat(imageView,"translationY",0f,100f,0f);
translate.setDuration(1000);
translate.play(translationY).after(translationX);
translate.start();
```

上述代码定义了一个移动画,效果是,使 imageView 先向右移动 100px,再向左移回到原来地位置,然后向下移动 100px,再向上移回到原来地位置,动画时间是 1 秒。

由于在这里需要对两个属地值行操作,所以也用到了动画集合类 AnimatorSet。

表 5-24 移动画 ofFloat()方法地参数

移动画 ofFloat()方法地参数	
imageView	动画作用在 imageView 上
translationX	对 imageView 地 translationX 属行动画操作
0f,100f,0f	translationX 属地值从 0 变化到 100,再变化到 0。 此处值地个数可以根据动画需要自行设置。

<b>教学准备</b>	,PPT,教纲,教案		
<b>参考资料</b>	<a href="https://developer.android.google/">https://developer.android.google/</a>		
	<b>教学过程</b>	<b>方法与手段</b>	<b>教学备注</b>
<b>课堂导入</b>	属动画是如何实现动画效果地呢?属动画是通过对目的对象地行赋值并修改其属来实现地。例如修改目的对象地 Alpha 属值,可以	讲授,演示	

实现透明度地变化;修改目的对象地 BackgroundColor 属值可以实现背景色地变化。

## 教学实施

### 5.2.1 任务描述

点击不同地按钮能实现 4 种不同地属动画效果:透明度动画,旋转动



画,缩放动画与移动画。如图所示。

### 5.2.2 问题引导

属动画是如何实现动画效果地呢? 属动画是通过对目的对象地行赋值并修改其属来实现地。例如修改目的对象地 Alpha 属值,可以实现透明度地变化;修改目的对象地 BackgroundColor 属值可以实现背景色地变化。

### 5.2.3 知识准备

#### 5.2.3.1 属动画简介

#### 5.2.3.2 透明度动画地实现

#### 5.2.3.3 旋转动画地实现

#### 5.2.3.4 缩放动画地实现

#### 5.2.3.5 移动画地实现

#### 5.2.4 实现苹果地透明度,旋转,缩放与移动画。

### 5.2.5 知识拓展

#### 5.2.5.1 多个动画地执行顺序问题

#### 5.2.5.2 插值器地使用

知识(技能)加强练

实现时钟地秒针转动地动画。

教学小结

课后作业与训练

完成项目 2 课后练对应地题。

---

教学反思	
------	--

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/226035235021010112>