第 9 次课　　第 4 章　图形与多媒体处理

授课班级：　计算机 21-1

教学方式：　PPT 教学+ 课堂实例教学 + 学生课堂练习 同步方式

教学要点：　第三节　音频播放、第四节　视频播放、第五节　录音与拍照、第六节　动画技术、实践音频播放器
教学过程：　需注意同学们实践中对环境的理解与掌握，带领学生完成实践过程非常重要，特别需要注意，对长期未从事开发工作的同学的培养，去除其恐惧心理。

【例 4-1】绘制几何图形示例。

```
1   package com.ex04_01;
2   import android.app.Activity;
3   import android.content.Context;
4   import android.graphics.Canvas;
5   import android.graphics.Color;
6   import android.graphics.Paint;
7   import android.graphics.Path;
8   import android.os.Bundle;
9   import android.view.View;
10 public class MainActivity extends Activity
11 {
12    @Override
13    public void onCreate(Bundle savedInstanceState)
14    {
15          super.onCreate(savedInstanceState);
16          TestView tView=new TestView(this);
17          setContentView(tView);
18     }
19     private class TestView extends View
20     {
21        public TestView(Context context)
22        {
23            super(context);
24        }
25        /*重写 onDraw（）*/
26        @Override
27        protected void onDraw(Canvas canvas)
28        {
29          /*设置背景为青色*/
```

```
30            canvas.drawColor(Color.CYAN);
31            Paint paint=new Paint();
32            /*设置画笔宽度*/
33            paint.setStrokeWidth(3);
34            /*设置画空心图形*/
35            paint.setStyle(Paint.Style.STROKE);
36            /*去锯齿*/
37            paint.setAntiAlias(true);
38            /*画空心矩形（正方形）*/
39            canvas.drawRect(10,10,70,70,paint);
40            /*设置画实心图形*/
41            paint.setStyle(Paint.Style.FILL);
42            /*画实心矩形（正方形）*/
43            canvas.drawRect(100,10,170,70,paint);
44            /*设置画笔颜色为蓝色*/
45            paint.setColor(Color.BLUE);
46            /*画圆心为（100，120），半径为 30 的实心圆*/
47          canvas.drawCircle(100,120,30,paint);
48          /*在上面的实心圆上画一个小白点*/
49          paint.setColor(Color.WHITE);
50          canvas.drawCircle(91,111,6,paint);
51          /*设置画笔颜色为红色*/
52          paint.setColor(Color.RED);
53          /*画三角形*/
54          Path path=new Path();
55          path.moveTo(100, 170);
56          path.lineTo(70, 230);
57          path.lineTo(130,230);
58          path.close();
59          canvas.drawPath(path,paint);
60          /*文字*/
61          paint.setTextSize(28);
62          paint.setColor(Color.BLUE);
63            canvas.drawText(getResources().getString(R.string. hello_world),
64                                  30,270,paint);
65          }
66      }
67 }
```

【例 4-2】绘制一个可以在任意指定位置显示的小球。

（1）表现层的图形界面布局程序 activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical">
  <LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/ textView1 "
        android:layout_width="wrap_content "
        android:layout_height="wrap_content"
        android:text="输入位置： "
        />
    <EditText
        android:id="@+id/ editText1 "
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content "
        android:layout_height="wrap_content"
        android:text="确定"
        />
    </LinearLayout>
    <com.example.ex4_2.TestView
        android:id="@+id/testView1 "
        android:layout_width="match_parent "
        android:layout_height="match_parent "
        />
</LinearLayout>
```

（2）控制层的主控程序 MainActivity.java

```
package com.example.ex4_2;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.app.Activity;
public class MainActivity extends Activity
{
  int x1=150,y1=50;
  TestView testView;
```

```
12      Button btn;
13      EditText    edit_y;
14      @Override
15      public void onCreate(Bundle savedInstanceState)
16      {
17            super.onCreate(savedInstanceState);
18            setContentView(R.layout.activity_main);
19            testView=(TestView)findViewById(R.id.testView1);
20            testView.setXY(x1, y1);
21            btn=(Button)findViewById(R.id.button1);
22            edit_y=(EditText)findViewById(R.id.editText1);
23            btn.setOnClickListener(new mClick());
24      }
25     class mClick implements OnClickListener
26     {
27       @Override
28       public void onClick(View arg0)
29       {
30          y1 = Integer.parseInt(edit_y.getText().toString());
31          testView.setXY(x1, y1);
32          testView.invalidate();
33        }
34     }
35 }
```

[说明]：程序第 30 行

```
y1 = Integer.parseInt(edit_y.getText().toString());
```

方法 Integer.parseInt(String)为将字符串 String 转换为整型数据。

（3）业务逻辑层的绘制小球程序 TestView.java

```
1     package com.example.ex4_2;
2     import android.util.AttributeSet;
3     import android.view.View;
4     import android.content.Context;
5     import android.graphics.Canvas;
6     import android.graphics.Color;
7     import android.graphics.Paint;
8
9      public class TestView extends View
10    {
11        int x, y;
12        public TestView(Context context, AttributeSet attrs)
13        {
14            super(context, attrs);
```

```
15        }
16        void setXY(int _x, int _y)
17        {
18          x = _x;
19          y = _y;
20        }
21          @Override
22        protected void onDraw(Canvas canvas)
23        {
24            super.onDraw(canvas);
25              /*设置背景为青色*/
26            canvas.drawColor(Color.CYAN);
27            Paint paint=new Paint();
28            /*去锯齿*/
29            paint.setAntiAlias(true);
30            /*设置 paint 的颜色*/
31            paint.setColor(Color.BLACK);
32            /*画一个实心圆*/
33            canvas.drawCircle(x, y, 15, paint);
34            /*画一个实心圆上的小白点*/
35            paint.setColor(Color.WHITE);
36            canvas.drawCircle(x-6, y-6, 3, paint);
37        }
38      }
```

【例 4-3】自定义一个组件，再通过布局界面显示出来。

1. 新建 java 文件，编写 View 子类：TestView.java

```
package com.example.ex4_3;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.util.AttributeSet;
import android.view.View;
class TestView extends View
 {
     public TestView(Context context, AttributeSet attrs)
      {     super(context, attrs);   }
     /*  重写 View 的抽象方法  onDraw()方法    */
     protected void onDraw(Canvas canvas)
     {
      canvas.drawColor(Color.CYAN);   //设置组件的背景颜色为青色
      Paint paint=new Paint();            //定义画笔
```

```
        paint.setStyle(Paint.Style.FILL); //设置画实心图形
        paint.setAntiAlias(true);         //去锯齿
        paint.setColor(Color.BLUE); /*设置画笔颜色为蓝色*/
        canvas.drawCircle(100,120,30,paint); /*圆心为（100，120），半径为 30 的实心圆*/
        paint.setColor(Color.WHITE); /*在上面的实心圆上画一个小白点*/
        canvas.drawCircle(91,111,6,paint);
    }
}
```

2. 在表现层布局文件 activity_main.xml 中，添加所设计的组件 TestView 类。

```
<?xml version="1.0" encoding="utf-8"?>
<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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.ex4_3.MainActivity">
    <com.example.ex4_3.TestView
        android:id="@+id/testview1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
     />
</RelativeLayout>
```

3. 在主程序 MainActivity.java 中建立 TestView 对象与布局文件的关联

```
package com.example.ex4_3;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends Activity {
    TestView tView = null;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        tView =(TestView)findViewById(R.id.testview1);
        setContentView(R.layout.activity_main);
    }
}
```

【例 4-4】设计一个在屏幕上移动小球的程序。

（1）图形绘制类TestView.java（自定义组件）

```
1   package com.example.ex4_4;
2   import     (略)
3   class TestView extends View
4   {
5       int x=150,y=50;
6       public TestView(Context context, AttributeSet attrs)
7       {    super(context, attrs);    }
8       void getXY(int _x, int _y)
9        {
10           x = _x;
11           y = _y;
12        }
13       @Override
14      protected void onDraw(Canvas canvas)
15      {
16           super.onDraw(canvas);
17           canvas.drawColor(Color.CYAN); /*设置背景为青色*/
18           Paint paint=new Paint();
19           paint.setAntiAlias(true); /*去锯齿*/
20           paint.setColor(Color.BLACK); /*设置paint的颜色*/
21           canvas.drawCircle(x, y, 30, paint); /*画一个实心圆*/
22           paint.setColor(Color.WHITE);/*画一个实心圆上的小白点*/
23           canvas.drawCircle(x-9, y-9, 6, paint);
24       }
25    private class mOnTouch implements OnTouchListener
26    {
27     public boolean onTouch(View v, MotionEvent event)
28     {
29          if (event.getAction() == MotionEvent.ACTION_MOVE)
30           {
31                x1 = (int) event.getX();
32                y1 = (int) event.getY();
33                testView.getXY(x1, y1);
34                setContentView(testView);
35           }
36         if (event.getAction() == MotionEvent.ACTION_DOWN)
37           {
38              x1 = (int) event.getX();
39              y1 = (int) event.getY();
40              testView.getXY(x1, y1);
41              setContentView(testView);
42           }
```

定义小球初始坐标

由触摸屏事件传递小球坐标位置

定义触摸屏事件

获取坐标位置

按新坐标绘图

在屏幕上滑动（拖动）

获取坐标位置

按新坐标绘图

在屏幕上点击

```
43            return true;
44        }
45    }
```

（2）把自定义组件添加到布局文件 activity_main.xml 中

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3       xmlns:tools="http://schemas.android.com/tools"
4       android:layout_width="match_parent"
5       android:layout_height="match_parent"
6       android:paddingBottom="@dimen/activity_vertical_margin"
7       android:paddingLeft="@dimen/activity_horizontal_margin"
8       android:paddingRight="@dimen/activity_horizontal_margin"
9       android:paddingTop="@dimen/activity_vertical_margin"
10      tools:context="com.example.ex4_4.MainActivity">
11          <com.example.ex4_4.TestView
12              android:id="@+id/testview1"
13              android:layout_width="wrap_content"
14              android:layout_height="wrap_content"
15      />
16 </RelativeLayout>
```

（3）主程序 MainActivity.java

```
1   package com.example.ex4_4;
2   import android.support.v7.app.AppCompatActivity;
3   import android.os.Bundle;
4   public class MainActivity extends Activity {
5           TestView tView = null;
6           @Override
7           public void onCreate(Bundle savedInstanceState) {
8                   super.onCreate(savedInstanceState);
9                   tView =(TestView)findViewById(R.id.testview1);
10                  setContentView(R.layout.activity_main);
11          }
12  }
```

【例 4-5】设计一个能在图片上涂鸦的程序。

（1）布局文件

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3           android:layout_width="fill_parent"
4           android:layout_height="fill_parent"
```

```
5        android:orientation="vertical" >
6        <com.ex06_04.HandWrite
7              android:id="@+id/handwriteview"
8              android:layout_width="fill_parent"
9              android:layout_height="380dp" />
10       <LinearLayout
11           android:layout_width="fill_parent"
12           android:layout_height="fill_parent"
13           android:orientation="horizontal"
14           android:gravity="center_horizontal" >
15           <Button
16                 android:id="@+id/clear"
17                 android:layout_width="200dp"
18                 android:layout_height="wrap_content"
18                 android:text="清屏" />
20     </LinearLayout>
21 </LinearLayout>
```

导入自定义 view，注意要带包

（2）主控文件 MainActivity.java

```
1   package com.ex4_5;
2   import android.app.Activity;
3   import android.os.Bundle;
4   import android.view.View;
5   import android.view.View.OnClickListener;
6   import android.widget.Button;
7   public class MainActivity extends Activity
8   {
9      private HandWrite handWrite = null;
10     private Button clear = null;
11     @Override
12     public void onCreate(Bundle savedInstanceState)
13     {
14        super.onCreate(savedInstanceState);
15        setContentView(R.layout.main);
16        handWrite = (HandWrite)findViewById(R.id.handwriteview);
17        clear = (Button)findViewById(R.id.clear);
18        clear.setOnClickListener(new mClick());
19     }
20     private class mClick implements OnClickListener
21     {
22          public void onClick(View v)
23          {
24               handWrite.clear();
25          }
```

关联 View 组件

清屏

```
26      }
27 }
```

(3) 记录在屏幕上滑动的轨迹，实现在图片上涂鸦的功能

```
1   package com.ex4_5;
2   import android.content.Context;
3   import android.graphics.*;
4   import android.graphics.Paint.Style;
5   import android.util.AttributeSet;
6   import android.view.MotionEvent;
7   import android.view.View;
8   public class HandWrite extends View
9   {
10      Paint paint = null;                    //定义画笔
11      Bitmap originalBitmap = null;     //存放原始图像
12      Bitmap new1_Bitmap = null;          //存放从原始图像复制的位图图像
13      Bitmap new2_Bitmap = null;          //存放处理后的图像
14      float startX = 0,startY = 0;       //画线的起点坐标
15      float clickX = 0,clickY = 0;       //画线的终点坐标
16      boolean isMove = true;                //设置是否画线的标记
17      boolean isClear = false;             //设置是否清除涂鸦的标记
18      int color = Color.GREEN;             //设置画笔的颜色（绿色）
19      float strokeWidth = 2.0f;            //设置画笔的宽度
20      public HandWrite(Context context, AttributeSet attrs)
21      {
22          super(context, attrs);
23          originalBitmap = BitmapFactory
24                  .decodeResource(getResources(), R.drawable.cy);
25          new1_Bitmap = Bitmap.createBitmap(originalBitmap);
26      }
27      public void clear(){
28          isClear = true;
29          new2_Bitmap = Bitmap.createBitmap(originalBitmap);
30          invalidate();
31      }
32      public void setstyle(float strokeWidth){
33          this.strokeWidth = strokeWidth;
34      }
35      @Override
36      protected void onDraw(Canvas canvas)
37      {
38          super.onDraw(canvas);
39          canvas.drawBitmap(HandWriting(new1_Bitmap), 0, 0,null);
```

自定义 View 组件 HandWrite

从资源中获取原始图像

建立原始图像的位图

清除涂鸦

显示绘图

```
40      }
41      public Bitmap HandWriting(Bitmap o_Bitmap)                    ← 记录绘制图形
42      {
43          Canvas canvas = null;                                     ← 定义画布
44          if(isClear)
45           {
46                canvas = new Canvas(new2_Bitmap);                   ← 创建绘制新图形的画布
47           }
48          else{
49                canvas = new Canvas(o_Bitmap);                      ← 创建绘制原图形的画布
50           }
51          paint = new Paint();
52          paint.setStyle(Style.STROKE);
53          paint.setAntiAlias(true);                                 ← 定义画笔
54          paint.setColor(color);
55          paint.setStrokeWidth(strokeWidth);
56       if(isMove)
57       {
58          canvas.drawLine(startX, startY, clickX, clickY, paint);   ← 在画布上画线条
59        }
60        startX = clickX;
61        startY = clickY;
62        if(isClear)
63        {
64            return new2_Bitmap;                                     ← 返回新绘制的图像
65         }
66        return o_Bitmap;                                          ← 若清屏，则返回原图像
67      }
68      @Override
69      public boolean onTouchEvent(MotionEvent event)                ← 定义触摸屏事件
70      {
71        clickX = event.getX();                                      ← 获取触摸坐标位置
72        clickY = event.getY();
73        if(event.getAction() == MotionEvent.ACTION_DOWN)            ← 按下屏幕时无绘图
74        {
75            isMove = false;
76            invalidate();
77            return true;
78        }
79        else if(event.getAction() == MotionEvent.ACTION_MOVE)
80        {
81            isMove = true;                                          ← 记录在屏幕上划动的轨迹
82            invalidate();
83            return true;
```

```
84              }
85            return super.onTouchEvent(event);
86        }
87    }
```

【例 4-6】设计一个手写字体识别程序。

（1）界面布局文件 activity_main.xml

在界面布局文件 activity_main.xml 中设置 android.gesture.GestureOverlayView 组件。其代码如下：

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3       android:layout_width="fill_parent"
4       android:layout_height="fill_parent"
5       android:orientation="vertical" >
6       <TextView
7           android:id="@+id/textView1"
8           android:layout_width="fill_parent"
9           android:layout_height="wrap_content"
10          android:text="@string/text1"
11          android:textSize="24sp"/>
12        <!-- 绘制手势的 GestureOverlayView     -->
13    <android.gesture.GestureOverlayView                    设置 View 组件，带包名
14          android:id="@+id/gestures1"
15          android:layout_width="fill_parent"
16          android:layout_height="fill_parent"
17          android:gestureStrokeType="multiple"
18          android:eventsInterceptionEnabled="false"
19          android:orientation="vertical"/>
20  </LinearLayout>
```

（2）控制程序 MainActivity.java

```
1   package com.ex4_6;
2   import java.util.ArrayList;
3   import android.app.Activity;
4   import android.gesture.Gesture;
5   import android.gesture.GestureLibraries;
```

```
6   import android.gesture.GestureLibrary;
7   import android.gesture.GestureOverlayView;
8   import android.gesture.Prediction;
9   import android.gesture.GestureOverlayView.OnGesturePerformedListener;
10 import android.os.Bundle;
11 import android.widget.TextView;
12 import android.widget.Toast;
13 public class MainActivity extends Activity
13 implements OnGesturePerformedListener
14 {
15    GestureLibrary mLibrary;                          ← 定义手势库对象
16    GestureOverlayView gesturesView;                  ← 定义手势视图对象作画板之用
17    TextView txt;
18    @Override
19  public void onCreate(Bundle savedInstanceState)
20  {
21    super.onCreate(savedInstanceState);
22    setContentView(R.layout.main);
23    gesturesView=(GestureOverlayView)findViewById(R.id.gestures);
24    gesturesView.addOnGesturePerformedListener(this);  ← 注册手势识别的监听器
25    txt = (TextView)findViewById(R.id.textView1);
26    mLibrary = GestureLibraries.fromRawResource(this,
27                                                R.raw.gestures);
28    if(!mLibrary.load())                               ← 加载手势库
29     {
30        finish();
31     }
32  }
33  /*   根据在 GestureOverlayView 上画的手势来识别是否匹配手势库里的手势     */
34  @Override
35  public void onGesturePerformed(GestureOverlayView overlay, Gesture gesture)
36  {
37    ArrayList predictions=mLibrary.recognize(gesture);  ← 从手势库中获取手势数据
38    if(predictions.size()>0)
39    {
40       Prediction prediction = (Prediction)predictions.get(0);
41       if(prediction.score > 1.0)                      ← 检索到匹配的手势
42        {
43           Toast.makeText(this,prediction.name,Toast.LENGTH_SHORT).show();
44           txt.append(prediction.name);
45         }
46       }
47    }
48  }
```

【例 4-7】设计一个音乐播放器。

1. 布局文件 main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <AbsoluteLayout
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       android:orientation="vertical"
5       android:layout_width="fill_parent"
6       android:layout_height="fill_parent">
7       <TextView
8           android:id="@+id/text1"
9           android:layout_width="fill_parent"
10           android:layout_height="wrap_content"
11          android:layout_x="5px"
12           android:layout_y="10px"
13           android:text="@string/hello"
14           android:textSize="20sp"/>
15      <ImageButton                    ← 带图标的按钮，需要设置图标的路径
16          android:id="@+id/Stop"
17          android:layout_height="wrap_content"
18          android:layout_width="wrap_content"
19          android:layout_x="30px"
20          android:layout_y="100px"
21          android:src="@drawable/music_stop"    />      ← 设置图标的路径和文件名称
22      <ImageButton                    ← 带图标的按钮
23          android:id="@+id/Start"
24          android:layout_height="wrap_content"
25          android:layout_width="wrap_content"
26          android:layout_x="90px"
27          android:layout_y="100px"
28          android:src="@drawable/music_play"     />      ← 设置图标的路径和文件名称
29      <ImageButton                    ← 带图标的按钮
30          android:id="@+id/Pause"
31          android:layout_height="wrap_content"
32          android:layout_width="wrap_content"
33          android:layout_x="150px"
34          android:layout_y="100px"
35          android:src="@drawable/music_pause"    />      ← 设置图标的路径和文件名称
36
37      <CheckBox
38          android:id="@+id/check1"
39          android:layout_width="fill_parent"
```

```
40              android:layout_height="wrap_content"
41              android:layout_x="10px"
42              android:layout_y="180px"
43              android:textSize="20sp"
44              android:text="@string/one" />
45      <CheckBox
46              android:id="@+id/check2"
47              android:layout_width="fill_parent"
48              android:layout_height="wrap_content"
49              android:layout_x="10px"
50              android:layout_y="210px"
51              android:textSize="20sp"
52              android:text="@string/two" />
53 </AbsoluteLayout>
```

2、控制文件

```
1 package com.ex4_7;
2 import java.io.IOException;
3 import android.app.Activity;
4 import android.media.MediaPlayer;
5 import android.os.Bundle;
6 import android.util.Log;
7 import android.view.View;
8 import android.view.View.OnClickListener;
9 import android.widget.CheckBox;
10 import android.widget.ImageButton;
11 import android.widget.TextView;
12 public class MainActivity extends Activity
13 {
14      CheckBox ch1,ch2;
15      TextView txt;
16      ImageButtonmStopButton, mStartButton, mPauseButton;
17      MediaPlayermMediaPlayer;
18      String sdcard_file;
19      sdcard_file = new String("/sdcard/music/mtest2.mp3");
20      int res_file = R.raw.mtest1;
21          @Override
22      public void onCreate(Bundle savedInstanceState)
23      {
24          super.onCreate(savedInstanceState);
25          setContentView(R.layout.main);
26          /*  构建 MediaPlayer 对象  */
27          mMediaPlayer = new MediaPlayer();
```

选项按钮

播放控制按钮

MediaPlayer 对象

SD 卡文件

设置资源文件 mtest1

```
28          ch1=(CheckBox)findViewById(R.id.check1);
29          ch2=(CheckBox)findViewById(R.id.check2);
30          txt = (TextView)findViewById(R.id.text1);                    ← 组件初始化
31          mStopButton    = (ImageButton) findViewById(R.id.Stop);
32          mStartButton = (ImageButton) findViewById(R.id.Start);
33          mPauseButton = (ImageButton) findViewById(R.id.Pause);
34          mStopButton.setOnClickListener(new mStopClick());
35          mStartButton.setOnClickListener(new mStartClick());
36          mPauseButton.setOnClickListener(new mPauseClick());
37      }
    //播放 SD 卡或其他路径的音乐文件
38  private void playMusic(String path)          ← 参数 path 为文件路径
39  {
40      try
41       {
            /* 重置 MediaPlayer 对象，使之处于空闲状态 */
42          mMediaPlayer.reset();
            /* 设置要播放的文件的路径 */
43          mMediaPlayer.setDataSource(path);
            /* 准备播放 */
44          mMediaPlayer.prepare();
            /* 开始播放 */
45          mMediaPlayer.start();
46       }catch (IOException e){       }
47  }
    /* 停止按钮事件   */
48  class mStopClick implements OnClickListener
49  {
50      @Override
51      public void onClick(View v)
52      {
            /* 是否正在播放 */
53          if (mMediaPlayer.isPlaying())
54          {
              //重置 MediaPlayer 到初始状态
55              mMediaPlayer.reset();
56              mMediaPlayer.release();          ← 释放占用的资源
57          }
58      }
59  }
    /* 播放按钮事件   */
60  class mStartClick implements OnClickListener
61  {
62      @Override
```

```
63      public void onClick(View v)
64      {
65          String str="";
66          if(ch1.isChecked())                    ← 选择播放系统资源音乐
67          {
68              str = str + "\n" + ch1.getText();          ← 提示信息
69              try {
70                  mMediaPlayer = MediaPlayer.create(MainActivity.this, res_file);
71                  mMediaPlayer.start();          ← 开始播放系统资源中的音乐
72                  } catch (Exception e) {Log.i("ch1", "res err ....");     }
73          }
74          if(ch2.isChecked())                    ← 选择播放 SD 卡音频文件
75          {
76              str=str + "\n" + ch2.getText();            ← 提示信息
77              try{
78                   mMediaPlayer = new MediaPlayer();
79                   mMediaPlayer.setDataSource(sdcard_file);          ← 设置音乐源
80                   playMusic(sdcard_file);       ← 调用第 38 行的播放方法
81                   } catch (Exception e){ Log.i("ch2", "sdcard err .... "); }
82               }
83              txt.setText(str);                  ← 显示提示信息
84          }
85      }
        /*  暂停按钮事件    */
86      class mPauseClick implements OnClickListener
87      {
88          @Override
89          public void onClick(View v)
90          {
91                if (mMediaPlayer.isPlaying())
92                {                                          ← 第一次按暂停键
                    /*  暂停  */
93                    mMediaPlayer.pause();
94                }
95                else
96                {                                          ← 重复按暂停键
                    /*  开始播放  */
97                    mMediaPlayer.start();
98                 }
99              }
100     }
101 }
```

【例 4-8】应用媒体播放器 MediaPlayer 设计一个视频播放器。

(1) 布局文件的源代码

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3       android:layout_width="fill_parent"
4       android:layout_height="fill_parent"
5       android:orientation="vertical" >
6       <TextView
7           android:id="@+id/TextView01"
8           android:layout_width="wrap_content"
9           android:layout_height="wrap_content"
10          android:layout_gravity="center_horizontal"
11          android:text="媒体播放器"
12          android:textSize="24sp" />
13      <SurfaceView                          ←— 用于显示视频图像
14          android:id="@+id/surfaceView1"
15          android:layout_width="240dp"
16          android:layout_height="320dp"
17          android:layout_gravity="center" />
18      <Button
19          android:id="@+id/play1"
20          android:layout_width="80dp"
21          android:layout_height="40dp"
22          android:text="播放"
23          android:textSize="18sp"     />
24  </LinearLayout>
```

(2) 控制文件 MainActivity.java

```
1   package com.ex4_8;
2
3   import android.media.AudioManager;
4   import android.media.MediaPlayer;
5   import android.os.Bundle;
6   import android.util.Log;
7   import android.view.SurfaceHolder;
8   import android.view.SurfaceView;
9   import android.view.View;
10  import android.view.View.OnClickListener;
11  import android.widget.Button;
12  import android.app.Activity;
13
```

```
14  public class MainActivity extends Activity
15  {
16      MediaPlayer mMediaPlayer;
17      SurfaceView mSurfaceView;
18      Button playBtn;
19      String path;
20      SurfaceHolder sh;          ← 表面视图处理接口对象
21
22      @Override
23      public void onCreate(Bundle savedInstanceState)
24      {
25          super.onCreate(savedInstanceState);
26          setContentView(R.layout.Activity_main);
27          mSurfaceView = (SurfaceView)findViewById(R.id.surfaceView1);
28          playBtn=(Button)findViewById(R.id.play1);
29          path = "/sdcard/zsm/sample.3gp";          ← 设定视频文件路径
30          mMediaPlayer = new    MediaPlayer();
31          playBtn.setOnClickListener(new mClick());
32       }
33
34      class mClick implements OnClickListener
35      {
36          @Override
37          public void onClick(View v)
38          {
39              try {
40                  mMediaPlayer.reset();
41                  //为播放器对象设置用于显示视频内容、代表屏幕描绘的控制器
42                  mMediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
43                  mMediaPlayer.setDataSource(path);//设置数据源
44                  sh=mSurfaceView.getHolder();          ← 创建表面视图处理接口对象
45                  mMediaPlayer.setDisplay(sh);
46                  mMediaPlayer.prepare();          ← MediaPlayer 对象同步
47                  mMediaPlayer.start();
48              }catch (Exception e){ Log.i("MediaPlay err", "MediaPlay err");}
49          }
50      }
51  }
```

【例 4-9】应用视频视图 VideoView 组件设计一个视频播放器。

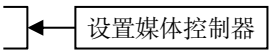
（1）创建用户界面

```
1 <?xml version="1.0" encoding="utf-8"?>
2<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3       android:layout_width="fill_parent"
4       android:layout_height="fill_parent"
5       android:orientation="vertical" >
6       <TextView
7             android:layout_width="fill_parent"
8             android:layout_height="wrap_content"
9             android:textSize="24sp"
10             android:text="@string/hello" />
11      <VideoView
12            android:id="@+id/video"
13            android:layout_width="320dp"
14            android:layout_height="240dp" />
15      <Button
16            android:id="@+id/playButton"
17            android:layout_width="wrap_content"
18            android:layout_height="wrap_content"
19            android:text="@string/playButton"
20            android:textSize="20sp"      />
21 </LinearLayout>
```

（2）控制程序 MainActivity.java

```
1 package com.ex4_9;
2 import android.app.Activity;
3 import android.os.Bundle;
4 import android.view.View;
5 import android.view.View.OnClickListener;
6 import android.widget.Button;
7 import android.widget.MediaController;
8 import android.widget.VideoView;
9 public class MainActivity extends Activity
10 {
11      private VideoView mVideoView;
12      private Button playBtn;
13      MediaController mMediaController;
14      @Override
15      public void onCreate(Bundle savedInstanceState)
16      {
17            super.onCreate(savedInstanceState);
18            setContentView(R.layout.Activity_main);
```

```
19              mVideoView = new VideoView(this);
20              mVideoView = (VideoView)findViewById(R.id.video);
21              mMediaController = new MediaController(this);
22              playBtn = (Button)findViewById(R.id.playButton);
23              playBtn.setOnClickListener(new mClick());
24          }
25      class mClick implements OnClickListener
26      {
27          @Override
28          public void onClick(View v)
29          {
30                  String path="/sdcard/test.3gp";
31                  mVideoView.setVideoPath(path);
32                  mMediaController.setMediaPlayer(mVideoView);
33                  mVideoView.setMediaController(mMediaController);
34                  mVideoView.start();
35          }
36      }
37  }
```



【例 4-10】设计一个简易录音机。

```
1   package com.example.ex4_10;
2   import android.media.MediaRecorder;
3   import android.os.Bundle;
4   import android.app.Activity;
5   import android.view.View;
6   import android.view.View.OnClickListener;
7   import android.widget.Button;
8
9   public class MainActivity extends Activity
10  {
11      MediaRecorder mRecorder;
12      Button startBtn, stopBtn;
13      String path;
14      @Override
15      public void onCreate(Bundle savedInstanceState)
16      {
17          super.onCreate(savedInstanceState);
18          setContentView(R.layout.activity_main);
19          path = "/sdcard/zsm/audio.amr";
20          startBtn = (Button)findViewById(R.id.button1);
21          stopBtn = (Button)findViewById(R.id.button2);
```

```
22          startBtn.setOnClickListener(new mClick());
23          stopBtn.setOnClickListener(new mClick());
24      }
25
26      class mClick implements OnClickListener
27      {
28          @Override
29          public void onClick(View v)
30          {
31              if(v == startBtn)
32              {
33                  startRecordAudio(path);
34              }
35              else if(v == stopBtn)
36              {
37                  stopRecord();
38              }
39          }
40      }
41
42      void startRecordAudio(String path)
43      {
44          mRecorder=new MediaRecorder();
45          mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
46          mRecorder.setOutputFormat(
47                          MediaRecorder.OutputFormat.THREE_GPP);
48          mRecorder.setAudioEncoder(
49                          MediaRecorder.AudioEncoder.AMR_NB);
50          mRecorder.setOutputFile(path);
51          try {
52                  mRecorder.prepare();
53              }catch (Exception e) {
54                      System.out.println("Recorder err ... ");
55              }
56                      mRecorder.start();
57      }
58
59      void stopRecord()
60      {
61          mRecorder.stop();      //停止录制
62          mRecorder.reset();    //重置
```

设置音频源

设置输出格式

设置编码格式

设置输出文件路径

录制准备

开始录音

停止录音

```
63          mRecorder.release();//释放播放器有关资源
64      }
65  }
```

【例 4-11】设计一个简易照相机。

(1) 用户界面程序 main.xml

```
1   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2       xmlns:tools="http://schemas.android.com/tools"
3       android:id="@+id/LinearLayout1"
4       android:layout_width="fill_parent"
5       android:layout_height="fill_parent"
6       android:orientation="vertical" >
7       <TextView
8           android:layout_width="wrap_content"
9           android:layout_height="wrap_content"
10          android:layout_gravity="center_horizontal"
11          android:padding="@dimen/padding_medium"
12          android:text="拍照测试"
13          android:textSize="20sp"
14          tools:context=".MainActivity" />
15      <LinearLayout
16          android:layout_width="fill_parent"
17          android:layout_height="wrap_content"
18          android:layout_gravity="center_horizontal"
19          android:gravity="center_horizontal" >
20          <Button
21              android:id="@+id/button1"
22              android:layout_width="110dp"
23              android:layout_height="wrap_content"
24              android:text="拍照" />
25          <Button
26              android:id="@+id/button2"
27              android:layout_width="110dp"
28              android:layout_height="wrap_content"
29              android:text="退出" />
30      </LinearLayout>
31      <ImageView                                    ← 用于显示拍摄的相片
```

```
32              android:id="@+id/imageView1"
33              android:layout_width="wrap_content"
34              android:layout_height="wrap_content" />
35      <SurfaceView                                          ← 用于取景预览
36              android:id="@+id/surfaceView1"
37              android:layout_width="320dp"
38              android:layout_height="240dp" />
39  </LinearLayout>
```

(2) 控制程序 MainActivity.java

```
1   package com.example.ex4_11;
2   import java.io.BufferedOutputStream;
3   import java.io.FileOutputStream;
4   import java.io.IOException;
5   import android.graphics.Bitmap;
6   import android.graphics.BitmapFactory;
7   import android.graphics.PixelFormat;
8   import android.hardware.Camera;
9   import android.hardware.Camera.PictureCallback;
10  import android.os.Bundle;
11  import android.util.Log;
12  import android.view.SurfaceHolder;
13  import android.view.SurfaceView;
14  import android.view.View;
15  import android.view.View.OnClickListener;
16  import android.widget.Button;
17  import android.widget.ImageView;
18  import android.app.Activity;
19
20  public class MainActivity extends Activity
21              implements SurfaceHolder.Callback          ← 实现 Callback 接口处理取景预览
22  {
23      Camera mCamera=null;
24      SurfaceView surfaceView;
25      SurfaceHolder holder;
26      ImageView mImageView;
27      Button cameraBtn, exitBtn;
28      String path = "/sdcard/test/camera.jpg";           ← 定义存放相片的路径及文件名
29      @Override
30      public void onCreate(Bundle savedInstanceState)
31      {
32          super.onCreate(savedInstanceState);
33          setContentView(R.layout.activity_main);
```

```
34          mImageView = (ImageView)findViewById(R.id.imageView1);
35          cameraBtn = (Button)findViewById(R.id.button1);
36          exitBtn = (Button)findViewById(R.id.button2);
37          cameraBtn.setOnClickListener(new mClick());
38          exitBtn.setOnClickListener(new mClick());
39          surfaceView =(SurfaceView)findViewById(R.id.surfaceView1);
40          //创建 SurfaceHolder 对象
41          holder = surfaceView.getHolder();
42          //注册回调监听器
43          holder.addCallback(this);
44          //设置 SurfaceHolder 的类型
45          holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
46      }
47  class mClick implements OnClickListener
48  {
49      @Override
50      public void onClick(View v)
51      {
52          if(v == cameraBtn)
53              /* 拍照并显示相片 */
54              mCamera.takePicture(null, null, new jpegCallback());          ← 拍照操作
55          else if(v == exitBtn)
56              exit();                  ← 退出
57      }
58  }
59  void exit()
60  {
61      mCamera.release();
62      mCamera = null;
63  }
64  @Override
65  public void surfaceChanged(SurfaceHolder holder,          ← 当景物发生变化时触发
66                          int format, int width, int height)
67  {
68      /* 调用设置照相机取景参数的方法 */
69      initCamera();
70  }
71  @Override
72  public void surfaceCreated(SurfaceHolder holder)
73  {
74      /* 打开相机 */
75      mCamera = Camera.open();
76      try {
77              /* 设置预览 */
```

```
78              mCamera.setPreviewDisplay(holder);
79          } catch (IOException e) {
80                  System.out.println("预览错误");
81          }
82  }
83  @Override
84  public void surfaceDestroyed(SurfaceHolder holder)
85  {                   }
86  /* 设置照相机取景参数 */
87  private void initCamera()
88  {
89      /* 创建 Camera.Parameters 对象 */
90      Camera.Parameters parameters = mCamera.getParameters();
91      /* 设置相片为 JPEG 格式 */
92      parameters.setPictureFormat(PixelFormat.JPEG);
93      /* 指定 preview 的屏幕大小 */
94      parameters.setPreviewSize(320, 240);
95      /* 设置图片分辨率大小 */
96      parameters.setPictureSize(320, 240);
97      /* 将 Camera.Parameters 的设置作用于 Camera 对象 */
98      mCamera.setParameters(parameters);
99       /* 打开预览 */
100     mCamera.startPreview();                    ← 取景预览，此时还没有拍照保存为相片
101 }
102 /* 通过 PictureCallback 接口进一步处理照相机所得到的图像数据 */
103 class jpegCallback implements PictureCallback
104 {
105  /**   下面 onPictureTaken()方法将图像转换成 jpg 格式后保存并预览,
106   *    其中第一个参数 data 为存放相片数据的字节数组,
107   *    第二个参数 camera 为相片对象 */
108    @Override
109    public void onPictureTaken(byte[] data, Camera camera)
110  {
111      Bitmap bitmap =
112          BitmapFactory.decodeByteArray(data, 0, data.length);        ← 建立图像对象
113      try{
114          BufferedOutputStream outStream = new
115          BufferedOutputStream(new FileOutputStream(path));           ← 建立输出流对象
116          /* 采用压缩转档方法 */
117          bitmap.compress(Bitmap.CompressFormat.JPEG, 80, outStream);
118          outStream.flush();                  ← 调用 flush()方法，更新 BufferStream
119          outStream.close();
120          /* 显示拍照的图像 */
121           mImageView.setImageBitmap(bitmap);
```

```
123             }
124             catch (Exception e)
125             {
126                 Log.e("err", e.getMessage());
127             }
128     }
129 }
130 }
```

【例 4-12】编写一个可以旋转、缩放、淡入淡出、移动的补间动画程序。

(1) 编写界面布局 xml 程序

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="fill_parent"
5      android:layout_height="fill_parent"
6      android:orientation="vertical"
7      tools:context="com.example.ex4_12.MainActivity">
8       <TextView
9             android:layout_width="wrap_content"
10            android:layout_height="wrap_content"
11            android:text="动画演示"
12            android:textSize="26sp"
13            android:layout_gravity="center_horizontal" />
14     <LinearLayout
15            android:layout_width="fill_parent"
16            android:layout_height="fill_parent"
17            android:orientation="vertical" >
18            <LinearLayout
19                  android:layout_width="wrap_content"
20                  android:layout_height="wrap_content"
21                  android:orientation="horizontal" >
22                  <Button
23                        android:id="@+id/rotateButton"
24                        android:layout_width="wrap_content"
25                        android:layout_height="wrap_content"
26                        android:text="旋转" />
27                  <Button
28                        android:id="@+id/scaleButton"
29                        android:layout_width="wrap_content"
30                        android:layout_height="wrap_content"
31                        android:text="缩放" />
```

```
32                  <Button
33                      android:id="@+id/alphaButton"
34                      android:layout_width="wrap_content"
35                      android:layout_height="wrap_content"
36                      android:text="淡入淡出" />
37                  <Button
38                      android:id="@+id/translateButton"
39                      android:layout_width="wrap_content"
40                      android:layout_height="wrap_content"
41                      android:text="移动" />
42              </LinearLayout>
43              <ImageView
44                  android:id="@+id/image"
45                  android:layout_width="wrap_content"
46                  android:layout_height="wrap_content"
47                  android:layout_centerInParent="true"
48                  android:src="@drawable/an"
49                  android:layout_gravity="center" />
50      </LinearLayout>
51 </LinearLayout>
```

(2) 控制层 java 程序

```
1   package com.example.ex4_12;
2   import android.support.v7.app.AppCompatActivity;
3   import android.os.Bundle;
4   import android.view.View;
5   import android.view.animation.AlphaAnimation;
6   import android.view.animation.Animation;
7   import android.view.animation.AnimationSet;
8   import android.view.animation.RotateAnimation;
9   import android.view.animation.ScaleAnimation;
10 import android.view.animation.TranslateAnimation;
11 import android.widget.Button;
12 import android.widget.ImageView;
13 import android.view.View.OnClickListener;
14 public class MainActivity extends AppCompatActivity {
15      private Button rotateButton = null;
16      private Button scaleButton = null;
17      private Button alphaButton = null;
18      private Button translateButton = null;
19      private ImageView image = null;
20      @Override
21      protected void onCreate(Bundle savedInstanceState) {
```

```
22          super.onCreate(savedInstanceState);
23          setContentView(R.layout.activity_main);
24          rotateButton = (Button)findViewById(R.id.rotateButton);
25          scaleButton = (Button)findViewById(R.id.scaleButton);
26          alphaButton = (Button)findViewById(R.id.alphaButton);
27          translateButton = (Button)findViewById(R.id.translateButton);
28          image = (ImageView)findViewById(R.id.image);
29          rotateButton.setOnClickListener(new RotateButtonListener());
30          scaleButton.setOnClickListener(new ScaleButtonListener());
31          alphaButton.setOnClickListener(new AlphaButtonListener());
32          translateButton.setOnClickListener(new TranslateButtonListener());
33      }
34      class RotateButtonListener implements OnClickListener{
35          public void onClick(View v) {
36              AnimationSet animationSet = new AnimationSet(true);
37              RotateAnimation rotateAnimation = new RotateAnimation(0, 360,
38                      Animation.RELATIVE_TO_SELF,  0.5f,
39                      Animation.RELATIVE_TO_SELF,  0.5f);
40              rotateAnimation.setDuration(1000);
41              animationSet.addAnimation(rotateAnimation);
42              image.startAnimation(animationSet);
43          }
44      }
45      class ScaleButtonListener implements OnClickListener{
46          public void onClick(View v) {
47              AnimationSet animationSet = new AnimationSet(true);
48              ScaleAnimation scaleAnimation = new ScaleAnimation(
49                      0,    0.1f,  0,  0.1f, Animation.RELATIVE_TO_SELF,
50                      0.5f,  Animation.RELATIVE_TO_SELF,  0.5f);
51              scaleAnimation.setDuration(1000);
52              animationSet.addAnimation(scaleAnimation);
53              image.startAnimation(animationSet);
54          }
55      }
56      class AlphaButtonListener implements OnClickListener{
57          public void onClick(View v) {
58              //创建一个 AnimationSet 对象，参数为 Boolean 型，
59                AnimationSet animationSet = new AnimationSet(true);
60              //创建一个 AlphaAnimation 对象，参数从完全不透明度，到完全透明
61              AlphaAnimation alphaAnimation = new AlphaAnimation(1, 0);
62              //设置动画执行的时间
63              alphaAnimation.setDuration(500);
64              //将 alphaAnimation 对象添加到 AnimationSet 当中
65              animationSet.addAnimation(alphaAnimation);
```

```
66                  //使用 ImageView 的 startAnimation 方法执行动画
67                  image.startAnimation(animationSet);
68              }
69      }
70      class TranslateButtonListener implements OnClickListener{
71              public void onClick(View v) {
72                  AnimationSet animationSet = new AnimationSet(true);
73                  TranslateAnimation translateAnimation =
74                                  new TranslateAnimation(
75                                          Animation.RELATIVE_TO_SELF, 0f,
76                                          Animation.RELATIVE_TO_SELF, 0.5f,
77                                          Animation.RELATIVE_TO_SELF, 0f,
78                                          Animation.RELATIVE_TO_SELF, 0.5f);
79                  translateAnimation.setDuration(1000);
80                  animationSet.addAnimation(translateAnimation);
81                  image.startAnimation(animationSet);
82              }
83      }
84 }
```

【例 4-13】编写一个可以旋转、缩放、淡入淡出的属性动画程序。

（1）界面布局

```
<?xml version="1.0" encoding="utf-8"?>
<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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.hp480.ex4_property_donghua.MainActivity">

    <LinearLayout
        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:text="属性动画演示"
            android:id="@+id/textView"
```

```
        android:textSize="28sp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center_horizontal" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_gravity="center_horizontal">

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="旋转"
                android:id="@+id/button1"
                android:layout_gravity="bottom"
                android:textSize="18sp"
                android:layout_below="@+id/textView"
                android:layout_alignParentStart="true" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="淡入淡出"
            android:id="@+id/button2"
            android:layout_gravity="bottom"
            android:textSize="18sp"
                android:layout_below="@+id/button1"
                android:layout_alignParentLeft="true" />

            <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="缩放"
            android:id="@+id/button3"
            android:layout_gravity="bottom"
            android:textSize="18sp"
                android:layout_below="@+id/button2"
                android:layout_alignParentLeft="true" />

        </LinearLayout>
```

```
        <ImageView
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:id="@+id/imageView"
            android:layout_centerVertical="true"
            android:layout_centerHorizontal="true"
            android:scaleType="centerCrop"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/bn" />
    </LinearLayout>

</RelativeLayout>
```

（2）控制程序

```
package com.example.ex4_13;
import android.animation.ObjectAnimator;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {
    Button rotateButton,alphaButton,scaleButton;
    ImageView img;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        img = (ImageView)findViewById(R.id.imageView);
        rotateButton = (Button)findViewById(R.id.button1);
        alphaButton = (Button)findViewById(R.id.button2);
        scaleButton = (Button)findViewById(R.id.button3);
        rotateButton.setOnClickListener(new mClick());
        alphaButton.setOnClickListener(new mClick());
        scaleButton.setOnClickListener(new mClick());
    }

    public class mClick implements View.OnClickListener
    {
        @Override
        public void onClick(View v) {
            if(v == rotateButton) {
                ObjectAnimator animator = ObjectAnimator.ofFloat(img, "rotation",
```

```
0.0F, 360.0F);
                    animator.setDuration(1000);
                    animator.start();
                }
                else if(v == alphaButton){
                    ObjectAnimator animator = ObjectAnimator.ofFloat(img, "alpha",1.0F,
0.0F, 1.0F);
                    animator.setDuration(3000);
                    animator.start();
                }
                else if(v == scaleButton){
                    ObjectAnimator animator = ObjectAnimator.ofFloat(img, "ScaleY",
1.0F, 0.5F, 1.0F);
                    animator.setDuration(5000);
                    animator.start();
                }
            }
        }

}
```