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。

```
package com.ex04_01;
import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Path;
import android.os.Bundle;
import android.view.View;
public class MainActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
          super.onCreate(savedInstanceState);
          TestView tView=new TestView(this);
          setContentView(tView);
     }
     private class TestView extends View
     {
        public TestView(Context context)
        {
            super(context);
        }
        /*重写 onDraw（）*/
        @Override
        protected void onDraw(Canvas canvas)
        {
          /*设置背景为青色*/
```

```
         canvas.drawColor(Color.CYAN);
         Paint paint=new Paint();
         /*设置画笔宽度*/
         paint.setStrokeWidth(3);
         /*设置画空心图形*/
         paint.setStyle(Paint.Style.STROKE);
         /*去锯齿*/
         paint.setAntiAlias(true);
         /*画空心矩形（正方形）*/
         canvas.drawRect(10,10,70,70,paint);
         /*设置画实心图形*/
         paint.setStyle(Paint.Style.FILL);
         /*画实心矩形（正方形）*/
         canvas.drawRect(100,10,170,70,paint);
         /*设置画笔颜色为蓝色*/
         paint.setColor(Color.BLUE);
         /*画圆心为（100，120），半径为30的实心圆*/
        canvas.drawCircle(100,120,30,paint);
        /*在上面的实心圆上画一个小白点*/
        paint.setColor(Color.WHITE);
        canvas.drawCircle(91,111,6,paint);
        /*设置画笔颜色为红色*/
        paint.setColor(Color.RED);
        /*画三角形*/
        Path path=new Path();
        path.moveTo(100, 170);
        path.lineTo(70, 230);
        path.lineTo(130,230);
        path.close();
        canvas.drawPath(path,paint);
        /*文字*/
        paint.setTextSize(28);
        paint.setColor(Color.BLUE);
         canvas.drawText(getResources().getString(R.string. hello_world),
                                 30,270,paint);
        }
    }
}
```

【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

```
package com.example.ex4_2;
import android.util.AttributeSet;
import android.view.View;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;

 public class TestView extends View
 {
     int x, y;
     public TestView(Context context, AttributeSet attrs)
     {
         super(context, attrs);
```

```
    }
    void setXY(int _x, int _y)
    {
      x = _x;
      y = _y;
    }
      @Override
    protected void onDraw(Canvas canvas)
    {
        super.onDraw(canvas);
          /*设置背景为青色*/
        canvas.drawColor(Color.CYAN);
        Paint paint=new Paint();
        /*去锯齿*/
        paint.setAntiAlias(true);
        /*设置 paint 的颜色*/
        paint.setColor(Color.BLACK);
        /*画一个实心圆*/
        canvas.drawCircle(x, y, 15, paint);
        /*画一个实心圆上的小白点*/
        paint.setColor(Color.WHITE);
        canvas.drawCircle(x-6, y-6, 3, paint);
    }
  }
```

【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）

```
package com.example.ex4_4;
import     (略)
class TestView extends View
{
      int x=150,y=50;                                       定义小球初始坐标
      public TestView(Context context, AttributeSet attrs)
      {   super(context, attrs);   }
      void getXY(int _x, int _y)
        {
              x = _x;                                         由触摸屏事件传递小球坐标位置
              y = _y;
          }
        @Override
      protected void onDraw(Canvas canvas)
      {
              super.onDraw(canvas);
              canvas.drawColor(Color.CYAN); /*设置背景为青色*/
              Paint paint=new Paint();
              paint.setAntiAlias(true); /*去锯齿*/
              paint.setColor(Color.BLACK); /*设置paint的颜色*/
              canvas.drawCircle(x, y, 30, paint); /*画一个实心圆*/
              paint.setColor(Color.WHITE);/*画一个实心圆上的小白点*/
              canvas.drawCircle(x-9, y-9, 6, paint);
        }
    private class mOnTouch implements OnTouchListener                 定义触摸屏事件
    {
      public boolean onTouch(View v, MotionEvent event)
      {
            if (event.getAction() == MotionEvent.ACTION_MOVE)
              {
                    x1 = (int) event.getX();              获取坐标位置
                    y1 = (int) event.getY();                                   在屏幕上滑动（拖动）
                    testView.getXY(x1, y1);              按新坐标绘图
                    setContentView(testView);
              }
          if (event.getAction() == MotionEvent.ACTION_DOWN)
              {
                  x1 = (int) event.getX();                获取坐标位置
                  y1 = (int) event.getY();                                     在屏幕上点击
                  testView.getXY(x1, y1);                按新坐标绘图
                  setContentView(testView);
              }
            return true;
        }
    }
```

（2）把自定义组件添加到布局文件 activity_main.xml 中

```
<?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_4.MainActivity">
        <com.example.ex4_4.TestView
              android:id="@+id/testview1"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
      />
</RelativeLayout>
```

添加自定义组件

（3）主程序 MainActivity.java

```
package com.example.ex4_4;
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 4-5】设计一个能在图片上涂鸦的程序。

（1）布局文件

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

```
   android:orientation="vertical" >
   <com.ex06_04.HandWrite
       android:id="@+id/handwriteview"
       android:layout_width="fill_parent"
       android:layout_height="380dp" />
   <LinearLayout
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:orientation="horizontal"
      android:gravity="center_horizontal" >
      <Button
          android:id="@+id/clear"
          android:layout_width="200dp"
          android:layout_height="wrap_content"
          android:text="清屏" />
  </LinearLayout>
</LinearLayout>
```

导入自定义 view，注意要带包

（2）主控文件 MainActivity.java

```
package com.ex4_5;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class MainActivity extends Activity
{
    private HandWrite handWrite = null;
    private Button clear = null;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        handWrite = (HandWrite)findViewById(R.id.handwriteview);
        clear = (Button)findViewById(R.id.clear);
        clear.setOnClickListener(new mClick());
    }
    private class mClick implements OnClickListener
    {
          public void onClick(View v)
          {
              handWrite.clear();
          }
```

关联 View 组件

清屏

```
    }
}
```

（3）记录在屏幕上滑动的轨迹，实现在图片上涂鸦的功能

```
package com.ex4_5；
import android.content.Context；
import android.graphics.*；
import android.graphics.Paint.Style；
import android.util.AttributeSet；
import android.view.MotionEvent；
import android.view.View；
public class HandWrite extends View                自定义 View 组件 HandWrite
{
    Paint paint = null；                    //定义画笔
    Bitmap originalBitmap = null；      //存放原始图像
    Bitmap new1_Bitmap = null；          //存放从原始图像复制的位图图像
    Bitmap new2_Bitmap = null；          //存放处理后的图像
    float startX = 0,startY = 0；      //画线的起点坐标
    float clickX = 0,clickY = 0；      //画线的终点坐标
    boolean isMove = true；                //设置是否画线的标记
    boolean isClear = false；            //设置是否清除涂鸦的标记
    int color = Color.GREEN；              //设置画笔的颜色（绿色）
    float strokeWidth = 2.0f；           //设置画笔的宽度
    public HandWrite(Context context, AttributeSet attrs)
    {
        super(context, attrs)；
        originalBitmap = BitmapFactory                      从资源中获取原始图像
                .decodeResource(getResources(), R.drawable.cy)；
        new1_Bitmap = Bitmap.createBitmap(originalBitmap)；      建立原始图像的位图
    }
    public void clear(){
        isClear = true；
        new2_Bitmap = Bitmap.createBitmap(originalBitmap)；       清除涂鸦
        invalidate()；
    }
    public void setstyle(float strokeWidth){
        this.strokeWidth = strokeWidth；
    }
    @Override
    protected void onDraw(Canvas canvas)
    {
        super.onDraw(canvas)；                                         显示绘图
        canvas.drawBitmap(HandWriting(new1_Bitmap), 0, 0,null)；
```

```
     }
     public Bitmap HandWriting(Bitmap o_Bitmap)          ← 记录绘制图形
     {
        Canvas canvas = null;          ← 定义画布
        if(isClear)
         {
            canvas = new Canvas(new2_Bitmap);          ← 创建绘制新图形的画布
         }
        else{
            canvas = new Canvas(o_Bitmap);          ← 创建绘制原图形的画布
         }
        paint = new Paint();
        paint.setStyle(Style.STROKE);
        paint.setAntiAlias(true);          ← 定义画笔
        paint.setColor(color);
        paint.setStrokeWidth(strokeWidth);
     if(isMove)
     {
       canvas.drawLine(startX, startY, clickX, clickY, paint);          ← 在画布上画线条
      }
      startX = clickX;
      startY = clickY;
      if(isClear)
      {
         return new2_Bitmap;          ← 返回新绘制的图像
       }
      return o_Bitmap;          ← 若清屏，则返回原图像
    }
    @Override
    public boolean onTouchEvent(MotionEvent event)          ← 定义触摸屏事件
    {
      clickX = event.getX();          ← 获取触摸坐标位置
      clickY = event.getY();
      if(event.getAction() == MotionEvent.ACTION_DOWN)          ← 按下屏幕时无绘图
      {
         isMove = false;
         invalidate();
         return true;
      }
      else if(event.getAction() == MotionEvent.ACTION_MOVE)
      {
         isMove = true;          ← 记录在屏
         invalidate();              幕上划动
         return true;               的轨迹
```

```
            }
        return super.onTouchEvent(event);
    }
}
```

【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：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical" >
  <TextView
    android:id="@+id/textView1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/text1"
    android:textSize="24sp"/>
   <!-- 绘制手势的 GestureOverlayView   -->
 <android.gesture.GestureOverlayView
    android:id="@+id/gestures1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gestureStrokeType="multiple"
    android:eventsInterceptionEnabled="false"
    android:orientation="vertical"/>
</LinearLayout>
```

设置 View 组件，带包名

（2）控制程序 MainActivity.java

```
package com.ex4_6;
import java.util.ArrayList;
import android.app.Activity;
import android.gesture.Gesture;
import android.gesture.GestureLibraries;
import android.gesture.GestureLibrary;
import android.gesture.GestureOverlayView;
import android.gesture.Prediction;
import android.gesture.GestureOverlayView.OnGesturePerformedListener;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends Activity
implements OnGesturePerformedListener
{
    GestureLibrary mLibrary;                ← 定义手势库对象
    GestureOverlayView gesturesView;             ← 定义手势视图对象作画板之用
    TextView txt;
    @Override
  public void onCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    gesturesView=(GestureOverlayView)findViewById(R.id.gestures);
    gesturesView.addOnGesturePerformedListener(this);     ← 注册手势识别的监听器
    txt = (TextView)findViewById(R.id.textView1);
    mLibrary = GestureLibraries.fromRawResource(this,
                                        R.raw.gestures);
    if(!mLibrary.load())
      {                                                     ← 加载手势库
          finish();
      }
  }
  /*   根据在 GestureOverlayView 上画的手势来识别是否匹配手势库里的手势     */
  @Override
  public void onGesturePerformed(GestureOverlayView overlay, Gesture gesture)
  {
    ArrayList predictions=mLibrary.recognize(gesture);   ← 从手势库中获取手势数据
    if(predictions.size()>0)
    {
        Prediction prediction = (Prediction)predictions.get(0);
        if(prediction.score > 1.0)         ← 检索到匹配的手势
          {
              Toast.makeText(this,prediction.name,Toast.LENGTH_SHORT).show();
              txt.append(prediction.name);
          }
      }
    }
  }
```

【例 4-7】设计一个音乐播放器。

1. 布局文件 main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
      xmlns:android="http://schemas.android.com/apk/res/android"
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">
      <TextView
          android:id="@+id/text1"
          android:layout_width="fill_parent"
            android:layout_height="wrap_content"
          android:layout_x="5px"
            android:layout_y="10px"
            android:text="@string/hello"
            android:textSize="20sp"/>
        <ImageButton                      带图标的按钮，需要设置图标的路径
            android:id="@+id/Stop"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_x="30px"
            android:layout_y="100px"
            android:src="@drawable/music_stop"    />        设置图标的路径和文件名称
        <ImageButton                        带图标的按钮
            android:id="@+id/Start"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_x="90px"
            android:layout_y="100px"
            android:src="@drawable/music_play"      />      设置图标的路径和文件名称
        <ImageButton                        带图标的按钮
            android:id="@+id/Pause"
            android:layout_height="wrap_content"
            android:layout_width="wrap_content"
            android:layout_x="150px"
            android:layout_y="100px"
            android:src="@drawable/music_pause"    />       设置图标的路径和文件名称

      <CheckBox
            android:id="@+id/check1"
            android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="10px"
        android:layout_y="180px"
        android:textSize="20sp"
        android:text="@string/one" />
    <CheckBox
        android:id="@+id/check2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_x="10px"
        android:layout_y="210px"
        android:textSize="20sp"
        android:text="@string/two" />
</AbsoluteLayout>
```

2、控制文件

```
package com.ex4_7;
import java.io.IOException;
import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.CheckBox;
import android.widget.ImageButton;
import android.widget.TextView;
public class MainActivity extends Activity
{
    CheckBox ch1,ch2;                                   选项按钮
    TextView txt;
    ImageButtonmStopButton, mStartButton, mPauseButton;         播放控制按钮
    MediaPlayermMediaPlayer;                            MediaPlayer 对象
    String sdcard_file;
    sdcard_file = new String("/sdcard/music/mtest2.mp3");       SD 卡文件
    int res_file = R.raw.mtest1;                        设置资源文件 mtest1
      @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        /*  构建 MediaPlayer 对象  */
        mMediaPlayer = new MediaPlayer();
```

```
     ch1=(CheckBox)findViewById(R.id.check1);
     ch2=(CheckBox)findViewById(R.id.check2);
     txt = (TextView)findViewById(R.id.text1);                    组件初始化
     mStopButton    = (ImageButton) findViewById(R.id.Stop);
     mStartButton = (ImageButton) findViewById(R.id.Start);
     mPauseButton = (ImageButton) findViewById(R.id.Pause);
     mStopButton.setOnClickListener(new mStopClick());
     mStartButton.setOnClickListener(new mStartClick());
     mPauseButton.setOnClickListener(new mPauseClick());
   }
 //播放 SD 卡或其他路径的音乐文件
 private void playMusic(String path)          参数 path 为文件路径
 {
   try
    {
      /*  重置 MediaPlayer 对象，使之处于空闲状态  */
      mMediaPlayer.reset();
      /*  设置要播放的文件的路径  */
      mMediaPlayer.setDataSource(path);
      /*  准备播放  */
      mMediaPlayer.prepare();
      /*  开始播放  */
      mMediaPlayer.start();
    }catch (IOException e){       }
 }
 /*  停止按钮事件    */
 class mStopClick implements OnClickListener
 {
   @Override
   public void onClick(View v)
   {
      /*  是否正在播放  */
     if (mMediaPlayer.isPlaying())
      {
        //重置 MediaPlayer 到初始状态
         mMediaPlayer.reset();
         mMediaPlayer.release();          释放占用的资源
      }
   }
 }
/*  播放按钮事件    */
class mStartClick implements OnClickListener
{
  @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                  {
                        /*  开始播放  */           ← 重复按暂停键
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

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="媒体播放器"
        android:textSize="24sp" />
    <SurfaceView
        android:id="@+id/surfaceView1"
        android:layout_width="240dp"
        android:layout_height="320dp"
        android:layout_gravity="center" />
    <Button
        android:id="@+id/play1"
        android:layout_width="80dp"
        android:layout_height="40dp"
        android:text="播放"
        android:textSize="18sp"    />
</LinearLayout>
```

用于显示视频图像

(2) 控制文件 MainActivity.java

```
package com.ex4_8;

import android.media.AudioManager;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.util.Log;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.app.Activity;

```

```
public class MainActivity extends Activity
{
    MediaPlayer mMediaPlayer;
    SurfaceView mSurfaceView;
    Button playBtn;
    String path;
    SurfaceHolder sh;          ← 表面视图处理接口对象

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.Activity_main);
            mSurfaceView = (SurfaceView)findViewById(R.id.surfaceView1);
            playBtn=(Button)findViewById(R.id.play1);
            path = "/sdcard/zsm/sample.3gp";          ← 设定视频文件路径
            mMediaPlayer = new    MediaPlayer();
            playBtn.setOnClickListener(new mClick());
      }

    class mClick implements OnClickListener
    {
          @Override
          public void onClick(View v)
          {
              try {
                        mMediaPlayer.reset();
                        //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
                      mMediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
                      mMediaPlayer.setDataSource(path);//设置数据源
                      sh=mSurfaceView.getHolder();          ← 创建表面视图处理接口对象
                      mMediaPlayer.setDisplay(sh);
                      mMediaPlayer.prepare();          ← MediaPlayer 对象同步
                      mMediaPlayer.start();
                  }catch (Exception e){ Log.i("MediaPlay err", "MediaPlay err");}
          }
      }
}
```

【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

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
     android:layout_width="fill_parent"
     android:layout_height="fill_parent"
     android:orientation="vertical" >
     <TextView
          android:layout_width="fill_parent"
          android:layout_height="wrap_content"
          android:textSize="24sp"
            android:text="@string/hello" />
       <VideoView
            android:id="@+id/video"
            android:layout_width="320dp"
            android:layout_height="240dp" />
       <Button
          android:id="@+id/playButton"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="@string/playButton"
          android:textSize="20sp"      />
</LinearLayout>
```

（2）控制程序 MainActivity.java

```
package com.ex4_9;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.VideoView;
public class MainActivity extends Activity
{
       private VideoView mVideoView;
       private Button playBtn;
       MediaController mMediaController;
       @Override
       public void onCreate(Bundle savedInstanceState)
       {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.Activity_main);
```

```
        mVideoView = new VideoView(this);
        mVideoView = (VideoView)findViewById(R.id.video);
        mMediaController = new MediaController(this);
        playBtn = (Button)findViewById(R.id.playButton);
        playBtn.setOnClickListener(new mClick());
    }
class mClick implements OnClickListener
{
    @Override
    public void onClick(View v)
    {
        String path="/sdcard/test.3gp";
        mVideoView.setVideoPath(path);
        mMediaController.setMediaPlayer(mVideoView);
        mVideoView.setMediaController(mMediaController);
        mVideoView.start();
    }
}
}
```

设置媒体控制器

【例 4-10】设计一个简易录音机。

```
package com.example.ex4_10;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity
{
    MediaRecorder mRecorder;
    Button startBtn, stopBtn;
    String path;
    @Override
   public void onCreate(Bundle savedInstanceState)
   {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);
     path = "/sdcard/zsm/audio.amr";
     startBtn = (Button)findViewById(R.id.button1);
     stopBtn = (Button)findViewById(R.id.button2);
```

```
        startBtn.setOnClickListener(new mClick());
        stopBtn.setOnClickListener(new mClick());
 }

class mClick implements OnClickListener
{
    @Override
    public void onClick(View v)
    {
        if(v == startBtn)
        {
          startRecordAudio(path);
        }
        else if(v == stopBtn)
        {
          stopRecord();
        }
    }
}

void startRecordAudio(String path)
{
    mRecorder=new MediaRecorder();
    mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);        ← 设置音频源

    mRecorder.setOutputFormat(
                  MediaRecorder.OutputFormat.THREE_GPP);        ← 设置输出格式

   mRecorder.setAudioEncoder(
              MediaRecorder.AudioEncoder.AMR_NB);        ← 设置编码格式

   mRecorder.setOutputFile(path);              ← 设置输出文件路径
   try {
          mRecorder.prepare();            ← 录制准备
        }catch (Exception e) {
             System.out.println("Recorder err ... ");
        }
             mRecorder.start();           ← 开始录音
}

void stopRecord()            ← 停止录音
{
    mRecorder.stop();    //停止录制
    mRecorder.reset();   //重置
```

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

【例 4-11】设计一个简易照相机。

（1）用户界面程序 main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:padding="@dimen/padding_medium"
        android:text="拍照测试"
        android:textSize="20sp"
        tools:context=".MainActivity" />
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:gravity="center_horizontal" >
        <Button
            android:id="@+id/button1"
            android:layout_width="110dp"
            android:layout_height="wrap_content"
            android:text="拍照" />
        <Button
            android:id="@+id/button2"
            android:layout_width="110dp"
            android:layout_height="wrap_content"
            android:text="退出" />
    </LinearLayout>
    <ImageView
```

用于显示拍摄的相片

```
            android:id="@+id/imageView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <SurfaceView                                    ◄── 用于取景预览
            android:id="@+id/surfaceView1"
            android:layout_width="320dp"
            android:layout_height="240dp" />
</LinearLayout>
```

（2）控制程序 MainActivity.java

```
package com.example.ex4_11；
import java.io.BufferedOutputStream；
import java.io.FileOutputStream；
import java.io.IOException；
import android.graphics.Bitmap；
import android.graphics.BitmapFactory；
import android.graphics.PixelFormat；
import android.hardware.Camera；
import android.hardware.Camera.PictureCallback；
import android.os.Bundle；
import android.util.Log;
import android.view.SurfaceHolder；
import android.view.SurfaceView；
import android.view.View；
import android.view.View.OnClickListener；
import android.widget.Button；
import android.widget.ImageView；
import android.app.Activity；

public class MainActivity extends Activity
            implements SurfaceHolder.Callback          ◄── 实现 Callback 接口处理取景预览
{
   Camera mCamera=null；
   SurfaceView surfaceView；
   SurfaceHolder holder；
   ImageView mImageView；
   Button cameraBtn, exitBtn；
   String path = "/sdcard/test/camera.jpg"；        ◄── 定义存放相片的路径及文件名
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.activity_main);
```

```
     mImageView = (ImageView)findViewById(R.id.imageView1);
     cameraBtn = (Button)findViewById(R.id.button1);
     exitBtn = (Button)findViewById(R.id.button2);
     cameraBtn.setOnClickListener(new mClick());
     exitBtn.setOnClickListener(new mClick());
     surfaceView =(SurfaceView)findViewById(R.id.surfaceView1);
     //创建 SurfaceHolder 对象
     holder = surfaceView.getHolder();
     //注册回调监听器
     holder.addCallback(this);
     //设置 SurfaceHolder 的类型
     holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
 }
class mClick implements OnClickListener
{
  @Override
  public void onClick(View v)
  {
    if(v == cameraBtn)
      /*  拍照并显示相片  */
      mCamera.takePicture(null, null, new jpegCallback());          ← 拍照操作
    else if(v == exitBtn)
      exit();              ← 退出
  }
}
void exit()
{
   mCamera.release();
   mCamera = null;
}
@Override
public void surfaceChanged(SurfaceHolder holder,              ← 当景物发生变化时触发
                int format, int width, int height)
{
   /*  调用设置照相机取景参数的方法  */
   initCamera();
}
@Override
public void surfaceCreated(SurfaceHolder holder)
{
    /*  打开相机  */
   mCamera = Camera.open();
   try {
        /*  设置预览  */
```

```
            mCamera.setPreviewDisplay(holder);
        } catch (IOException e) {
                System.out.println("预览错误");
        }
}
@Override
public void surfaceDestroyed(SurfaceHolder holder)
{               }
/* 设置照相机取景参数 */
private void initCamera()
{
        /* 创建 Camera.Parameters 对象 */
        Camera.Parameters parameters = mCamera.getParameters();
        /* 设置相片为 JPEG 格式 */
        parameters.setPictureFormat(PixelFormat.JPEG);
        /* 指定 preview 的屏幕大小 */
        parameters.setPreviewSize(320, 240);
        /* 设置图片分辨率大小 */
        parameters.setPictureSize(320, 240);
        /* 将 Camera.Parameters 的设置作用于 Camera 对象 */
        mCamera.setParameters(parameters);
          /* 打开预览 */
        mCamera.startPreview();          ← 取景预览，此时还没有拍照保存为相片
  }
  /* 通过 PictureCallback 接口进一步处理照相机所得到的图像数据 */
  class jpegCallback implements PictureCallback
  {
    /**  下面 onPictureTaken()方法将图像转换成 jpg 格式后保存并预览，
      *    其中第一个参数 data 为存放相片数据的字节数组，
      *    第二个参数 camera 为相片对象 */
      @Override
      public void onPictureTaken(byte[] data, Camera camera)
    {
          Bitmap bitmap =
              BitmapFactory.decodeByteArray(data, 0, data.length);          ← 建立图像对象
          try{
                BufferedOutputStream outStream = new
                BufferedOutputStream(new FileOutputStream(path));          ← 建立输出流对象
                /* 采用压缩转档方法 */
                bitmap.compress(Bitmap.CompressFormat.JPEG, 80, outStream);
                outStream.flush();          ← 调用 flush()方法，更新 BufferStream
                outStream.close();
                /* 显示拍照的图像 */
                  mImageView.setImageBitmap(bitmap);
                }
                catch (Exception e)
                {
                    Log.e("err", e.getMessage());
                }
        }
    }
}
```

【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

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    tools:context="com.example.ex4_12.MainActivity">
      <TextView
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="动画演示"
              android:textSize="26sp"
              android:layout_gravity="center_horizontal" />
      <LinearLayout
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:orientation="vertical" >
              <LinearLayout
                      android:layout_width="wrap_content"
                      android:layout_height="wrap_content"
                      android:orientation="horizontal" >
                      <Button
                              android:id="@+id/rotateButton"
                              android:layout_width="wrap_content"
                              android:layout_height="wrap_content"
                              android:text="旋转" />
                      <Button
                              android:id="@+id/scaleButton"
                              android:layout_width="wrap_content"
                              android:layout_height="wrap_content"
                              android:text="缩放" />
```

```
            <Button
                android:id="@+id/alphaButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="淡入淡出" />
            <Button
                android:id="@+id/translateButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="移动" />
        </LinearLayout>
        <ImageView
            android:id="@+id/image"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:src="@drawable/an"
            android:layout_gravity="center" />
    </LinearLayout>
</LinearLayout>
```

（2）控制层 java 程序

```
package com.example.ex4_12;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.view.animation.Animation;
import android.view.animation.AnimationSet;
import android.view.animation.RotateAnimation;
import android.view.animation.ScaleAnimation;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.ImageView;
import android.view.View.OnClickListener;
public class MainActivity extends AppCompatActivity {
    private Button rotateButton = null;
    private Button scaleButton = null;
    private Button alphaButton = null;
    private Button translateButton = null;
    private ImageView image = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rotateButton = (Button)findViewById(R.id.rotateButton);
        scaleButton = (Button)findViewById(R.id.scaleButton);
        alphaButton = (Button)findViewById(R.id.alphaButton);
        translateButton = (Button)findViewById(R.id.translateButton);
        image = (ImageView)findViewById(R.id.image);
        rotateButton.setOnClickListener(new RotateButtonListener());
        scaleButton.setOnClickListener(new ScaleButtonListener());
        alphaButton.setOnClickListener(new AlphaButtonListener());
        translateButton.setOnClickListener(new TranslateButtonListener());
    }
    class RotateButtonListener implements OnClickListener{
        public void onClick(View v) {
            AnimationSet animationSet = new AnimationSet(true);
            RotateAnimation rotateAnimation = new RotateAnimation(0, 360,
                    Animation.RELATIVE_TO_SELF,   0.5f,
                    Animation.RELATIVE_TO_SELF,   0.5f);
            rotateAnimation.setDuration(1000);
            animationSet.addAnimation(rotateAnimation);
            image.startAnimation(animationSet);
        }
    }
    class ScaleButtonListener implements OnClickListener{
        public void onClick(View v) {
            AnimationSet animationSet = new AnimationSet(true);
            ScaleAnimation scaleAnimation = new ScaleAnimation(
                   0,    0.1f,   0,   0.1f, Animation.RELATIVE_TO_SELF,
                   0.5f,   Animation.RELATIVE_TO_SELF,   0.5f);
            scaleAnimation.setDuration(1000);
            animationSet.addAnimation(scaleAnimation);
            image.startAnimation(animationSet);
        }
    }
    class AlphaButtonListener implements OnClickListener{
        public void onClick(View v) {
            //创建一个 AnimationSet 对象，参数为 Boolean 型，
             AnimationSet animationSet = new AnimationSet(true);
            //创建一个 AlphaAnimation 对象，参数从完全不透明度，到完全透明
            AlphaAnimation alphaAnimation = new AlphaAnimation(1, 0);
            //设置动画执行的时间
            alphaAnimation.setDuration(500);
            //将 alphaAnimation 对象添加到 AnimationSet 当中
            animationSet.addAnimation(alphaAnimation);
            //使用 ImageView 的 startAnimation 方法执行动画
            image.startAnimation(animationSet);
        }
    }
    class TranslateButtonListener implements OnClickListener{
        public void onClick(View v) {
            AnimationSet animationSet = new AnimationSet(true);
            TranslateAnimation translateAnimation =
                    new TranslateAnimation(
                            Animation.RELATIVE_TO_SELF, 0f,
                            Animation.RELATIVE_TO_SELF, 0.5f,
                            Animation.RELATIVE_TO_SELF, 0f,
                            Animation.RELATIVE_TO_SELF, 0.5f);
            translateAnimation.setDuration(1000);
            animationSet.addAnimation(translateAnimation);
            image.startAnimation(animationSet);
        }
    }
}
```

【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();
            }
        }
    }

}
```